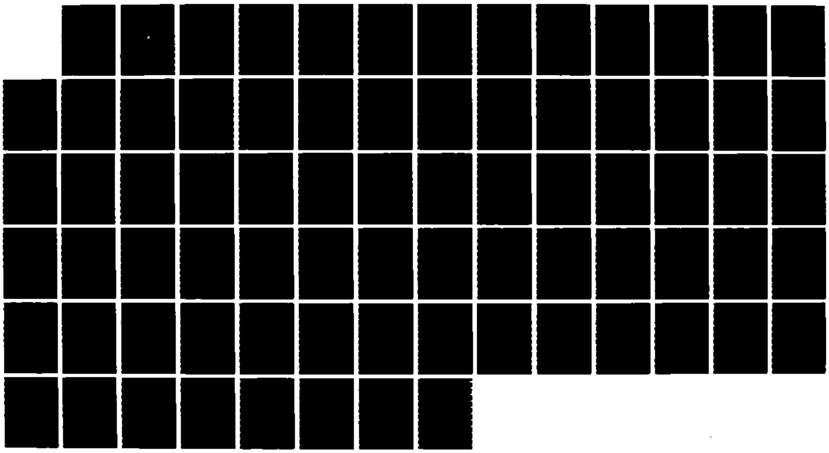
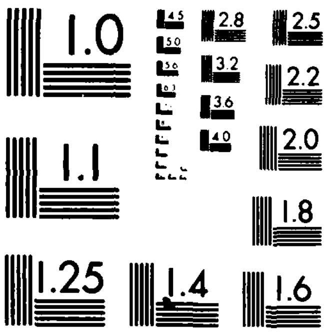
AD-A172 485 SATELLITE ORBIT PROGRAMS UTILIZING THE GRAPHICS CAPABILITIES OF THE MICROCOMPUTER(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA K A LANGDORF JUN 86 1/1

UNCLASSIFIED F/G 22/3 NL




1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SATELLITE ORBIT PROGRAMS UTILIZING THE GRAPHICS CAPABILITIES OF THE MICROCOMPUTER

by

Kim Alldredge Langdorf

June 1986

Thesis Advisor: G. L. Swafford

AD-A172485

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 39
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5001
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Satellite Orbit Programs Utilizing the Graphics Capabilities of the Microcomputer

12 PERSONAL AUTHOR(S)
Langdorf, Kim A.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1986 June	74

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial Satellite Orbits, Computer Graphics, Microcomputers

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

It is important for a student of Space Science to have the opportunity to thoroughly understand the principle of an artificial satellite orbit. This thesis consists of four computer graphics programs that will enable the student to see what an orbit is and how it works. The first program demonstrates the shape of an orbit in two dimensions resulting from initial altitude, speed, and flight path angle. The second program draws an orbit in three dimensions around a sphere based on the input of the classical orbital elements. The third program traces the ground track of a satellite over a map of the earth. And the fourth plots the ground track of a geosynchronous satellite over a map of the earth. The student can learn about orbits by entering the orbital elements and viewing the resultant orbit.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL G. L. Swafford	22b TELEPHONE (Include Area Code) (408) 646-2232 — 22c OFFICE SYMBOL 61Sw

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Satellite Orbit Programs Utilizing the Graphics Capabilities of the Microcomputer

by

Kim Alldredge Langdorf
Captain, United States Army
B.A., University of Utah, 1977

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Space Systems Operations)

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

Author: [illegible]
Kim Alldredge Langdorf

Approved by: [illegible]
G. L. Swafford, Thesis Advisor

Otto Heinz
Otto Heinz, Second Reader

[illegible]
Allen E. Fuhs, Chairman, Space Systems Academic Committee

J.N. Dyer, Dean of Science and Engineering

ABSTRACT

It is important for a student of Space Science to have the opportunity to thoroughly understand the principle of an artificial satellite orbit. This thesis consists of four computer graphics programs that will enable the student to see what an orbit is and how it works. The first program demonstrates the shape of an orbit in two dimensions resulting from initial altitude, speed, and flight path angle. The second program draws an orbit in three dimensions around a sphere based on the input of the classical orbital elements. The third program traces the ground track of a satellite over a map of the earth. And the fourth plots the ground track of a geosynchronous satellite over a map of the earth. The student can learn about orbits by entering the orbital elements and viewing the resultant orbit.

TABLE OF CONTENTS

List of Figures

List of Trademarks

Apple is a registered trademark of Apple Computer, Inc.
Atari is a registered trademark of Atari, Inc.
Microsoft is a trademark of Microsoft Corporation.
North Star is a trademark of North Star Computers, Inc.
TI is a trademark of Texas Instruments, Inc.
TRS-80 is a trademark of Tandy Corporation.
IBM is a registered trademark of International Business Machines, Inc.
The Amiga is manufactured by Commodore Business Machines, Inc.
Compaq is a trademark of Compaq Computer Corporation.

I. INTRODUCTION

In Principia Mathematica Sir Isaac Newton theorized that by firing a cannonball from the top of a high mountain at increasing speeds it would travel farther and farther until at last it would make a complete orbit around the earth [Ref. 1:p. 66]. As with many great discoveries, the idea of orbital flight came from a simple idea. Three hundred years later, on 4 October 1957, the launch of the first artificial satellite around the earth demonstrated that Newton's theory was essentially correct.

Spaceflight has always been tied closely to computers. Graphics is defined by the Oxford English Dictionary as "of drawing, painting, engraving, etching,etc.; vividly descriptive, lifelike; of diagrams and symbolic curves." Traditionally computer graphics evolved as a means of interpreting and displaying the increasing volume of numbers which the modern computer can manipulate with tremendous rapidity and accuracy. In 1963 Ivan Sutherland developed the first interactive computer graphics system called the "Sketchpad".
This advance in computer graphics enabled a user to interact directly with a computer graphics program in real time. [Ref. 2:p. vii]

A. OBJECTIVE

The laws of celestial mechanics as formulated by Tycho Brahe, Kepler, and Sir Isaac Newton came from simple ideas and meticulous observations of the heavens above. The intent of this thesis is to use the graphics capabilities of the microcomputer to help students of space science understand the simple laws of orbital mechanics that pertain to artificial satellites. It is sometimes easier to understand a scientific principle by first observing a demonstration of what it does and then studying all the theories, equations, formulas, and laws that explain the principle than by trying to visualize what it does from studying the equations alone.

Four software programs are the result of this effort:

1. INITIAL CONDITIONS demonstrates the shape of the orbit in two dimensions after the user enters the initial altitude, the initial flight path angle, and either the initial speed or the desired apogee height of the satellite.

2. ORBIT plots an orbit around a sphere representing the earth after the user inputs the classical orbital elements.

3. GROUND TRACK traces the ground track of an orbit on a modified cartesian projection map of the earth.

4. GEOSYNC displays the ground track of a geosynchronous orbit of any inclination or longitude of the equator crossing on a modified cartesian projection map of the earth.

B. HARDWARE

The decision to use a microcomputer for this project was based on the idea that microcomputers are more accessible and easier for the average student to use than are either minicomputers or mainframes. Until fairly recently the practice of computer graphics was restricted to special research laboratories and large scientific computing facilities. This was because the equipment necessary for computer graphics was expensive and bulky in size. In the mid-1970's came the flashing lights of the video arcades and inexpensive home video games. Several years later computer graphics became more available as the microcomputer revolution erupted. Apple computer offered the first microcomputer that could be operated with a minimum of fuss. And it had the added advantage of several types of color graphics. TRS-80, PET, Northstar, Heathkit, and in the late 1970's, Atari and Texas Instrument improved on what the Apple could do with graphics. In the summer of 1981 IBM entered the competition with the IBM PC. With Microsoft supplying the software IBM offered a microcomputer that quickly carved out a large share of the market. [Ref. 3: pp. 6-8]

In order for a computer to be useful as a learning tool, it must be readily available and easy to operate. The IBM Personal Computer (PC) was selected for this project for those reasons. It is one of the more popular PC's and is widely emulated by other computer manufacturers. A major portion of the programming for this project was done on a Compaq computer, a popular IBM compatible.

A typical *microcomputer* or PC is configured with a central processing unit (CPU) which is the "brain" of the computer, a keyboard which is the means to input user intructions to the CPU, and a screen (CRT) which provides the output to the user. In order to support computer graphics the PC must have a color/graphics adapter that produces both the color and the graphics on the screen. To store the programs and the software used to develop the programs a disk drive or other memory storage device is necessary. It is preferable to have two disk drives to cut down on the number of times a floppy disk has to be inserted and removed. The last recommended hardware component is a printer. It is very helpful to print out a program in order to debug it. And with the proper graphics software and printer it is possible to print the graphics output from the screen to the printer.

C. SOFTWARE

The software that controls the operation of a computer is called the operating system. It directs such operations as loading software to the internal memory and saving programs to the floppy disks. PC-DOS is one of the operating systems available for the IBM PC. It was chosen for this project because is offers the computer language BASICA. IBM BASICA was developed by Microsoft, Inc. and was derived from BASIC which was developed at Dartmouth College to be a computer language that is easy to learn and easy to work with. It is normally an interpreted language and that means that it is edited and programs are run in the same working environment. This makes BASIC a very user-friendly computer language. The disadvantage is that it is relatively slow and graphics can take a lot of computational time before being displayed. Compilers are available to speed things up by taking the source code and putting it in a form the computer can handle and display much faster.

IBM's BASICA offers some of the best graphics commands available on a microcomputer. The Graphics Definition Language (GDL) along with the LINE, CIRCLE, PSET, and PRESET commands provide powerful tools for graphics programming.

D. PROGRAM DESIGN

The general form of any software program is to accept user input from the keyboard, make the necessary calculations, and then output the results. In designing these programs an attempt was made to make them as user-friendly as possible. To start the programs, the user inserts the disk in the "A" or default drive and turns on the computer. The INTRO program is automatically loaded and run by a program called AUTOEXEC.BAT. If a different version of DOS is used, it is necessary to make sure that the BASICA is version 2.1 or higher. It might then be necessary to type BASICA at the system prompt and then 'load"INTRO",r' to start the programs.

BASICA has several features to help set up a graphics screen. The medium resolution screen is 320 by 200 picture elements (pixels). In addition to a background color there are three colors that can be used for drawing on the medium resolution screen. A higher resolution screen is available but it will not support color graphics. A portion of the screen can be used as a "window" by using the WINDOW and VIEWPORT commands. This allows many graphics functions such as scaling the size of objects, clipping objects that are too big for the window, and clearing only the screen inside the window.

Once the screen is set up, the user input is entered from the keyboard in response to prompts from the computer program. The entry is usually a number or a one or two letter response. The number can be entered in either real, integer, or floating point format. A yes or no response can be entered by typing a "y" or a "n". To select a program from the menu, type the first two letters in the name. In most cases the program will do a limited check to see if the input is appropriate. If not, a beep will sound and the question is again asked.

After the initial input is accepted, the program will do the calculations for the orbital data. The program then enters a loop that plots the orbit from the perigee point through one complete orbit around the earth and back to the perigee point. The orbit is plotted in steps of 2 time pi divided by an integer, i.e., 2 times pi divided by 50. After the loop is completed, the program will offer the user a choice of drawing another orbit or returning to the menu. Depending on the program, the orbital data is either given as the orbit is being plotted or as a viewing option after the orbit is drawn.

The physics formulas and equations in the programs came from a variety of sources. The BASICA language allows the programmer to enter an equation in the program in pretty much the same form as it is written in a textbook. Several exceptions to standard notation include using "*" for the multiplication operation, sqr( ) for the square root, and expressing exponentials as "X^2" for X to the second power. The equations in this thesis are written in the BASIC algorithmic language. The reason for doing so is to make the program listings as easy as possible to understand. The equations in the descriptions of the programs might not exactly match the equations in the listings because of special programming techniques which must be included in most computer programs to handle such problems as "division by zero". Another reason is that only a few trigonometric functions are directly computed in BASIC and the rest must be derived from those.

There are several common assumptions and constants used throughout the programs:

1. Both the earth and the satellite are considered to be spherically symmetric. This allows these bodies to be treated as though their masses are concentrated at their centers (point masses).
2. There are no other forces acting on the bodies other than the gravitational forces which act along a line joining the centers of the earth and the satellite.
3. Equatorial radius of the earth = 6378.165 kilometers
4. Gravitational constant of the earth = k = GMe = 398603.2 km^3/sec^2 [Ref. 4:pp. 11-12, Ref. 5:p. 113, Ref. 6:p. 374]

II. PROGRAM INTRO

This program displays the title page, introduction, and lists the menu of programs. It also ties all the programs together so that they can be called and run from the menu. Program INTRO is automatically loaded and run when the disk is inserted in the "A" or default drive and the computer is turned on. If the computer is already on, the system can be rebooted by pressing the Cntr, the Alt, and the Del keys at the same time, and releasing them together.

After the title page and a brief explanation of the programs, the user can run any of the programs by typing in the first two letters of the program name. The INTRO program will then load and run the chosen program. After the program is completed, the user has the option of running it again or returning to the menu for another choice of programs.

When the user wants to terminate using the programs, the only precaution is to make sure that the red light on the disk drive is out, indicating that the drive is not in use. The disk can then be taken out of the disk drive or the computer can be turned off.

III. PROGRAM INITIAL CONDITIONS

This program demonstrates the physics of the two-body problem. A similar program was written by Dr. John N. Dyer, Distinguished Professor of Physics, Naval Postgraduate School, for an early model Hewlett-Packard computer. The idea for this program and the majority of the equations come from Dr. Dyer's program; any equations used that were not from this source will be referenced. After an introduction, the user is prompted to enter the altitude above the surface of the earth of the satellite in kilometers. The altitude of the satellite is added to the radius of the earth to get the initial altitude of the satellite (Ro). The program then asks for the initial flight path angle in degrees (Bo), which is the angle between the velocity vector and the local horizontal plane. The final user input for the program is the selection of the speed in kilometers per second (Vo), or the desired apogee altitude above the surface of the earth in kilometers. If the user selects the apogee altitude, the program calculates the initial speed (Vo) as follows:

```
X = Ro/RAP

XXX = ((2-X-X^2*cos(Bo)^2-sqr((2-X-X^2*cos(Bo)^2)^2 4
      *(1-X)*(1-X^2*cos(Bo)^2)))/(1-X^2*cos(Bo)^2)

Vo = sqr((XXX*K)/Ro)
```

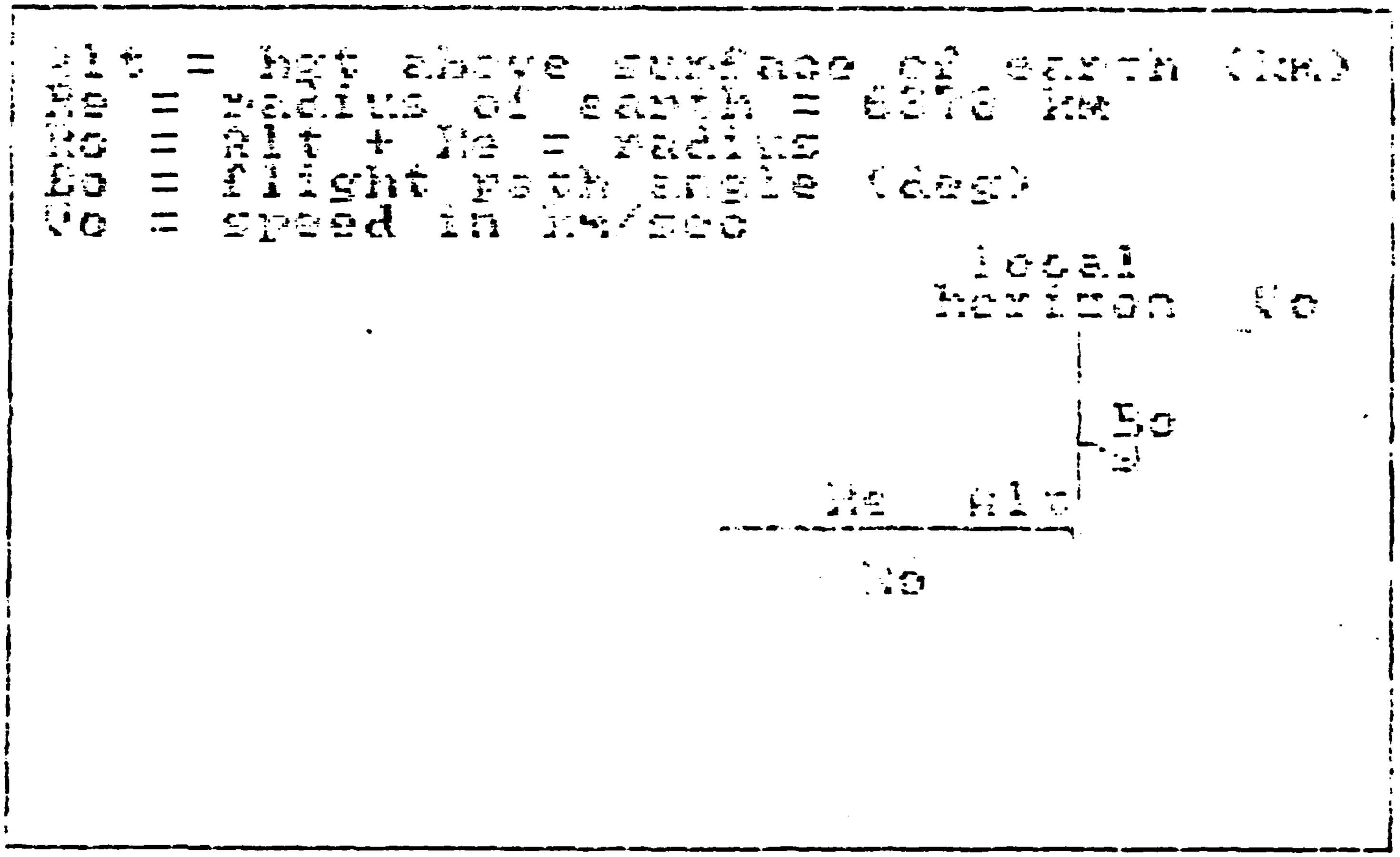

Alt = hgt above surface of earth (km)
Re = radius of earth = 6378 km
Ro = Alt + Re = radius
Bo = flight path angle (deg)
Vo = speed in km/sec

Figure 1. Initial conditions of an orbit

The program uses the initial conditions to find the specific angular momentum (SAM) in km^2 per second

SAM = Ro * Vo * cos(Bo) [Ref. 8:p. 74]

The specific mechanical energy (SME) in km^2 per sec^2 is

SME = (Vo^2/2)-(k/Ro) [Ref. 8:p. 74]

The shape of the orbit is determined by the eccentricity. The equation for eccentricity is

AAA = (Ro*Vo^2)/k

ECC = sqr((AAA-1)^2*(cos(Bo))^2+(sin(Bo))^2)

If the eccentricity = 0, the orbit is a circle.

If the eccentricity is between 0 and 1, the orbit is an ellipse.

If the eccentricity = 1, the orbit is a parabola.

If the eccentricity is greater than 1, the orbit is a hyperbola.

If the eccentricity is less than 0, the orbit is an ellipse with the apogee at the initial condition point.

Two velocities of interest are calculated next. They are the circular orbit speed (VCIR), and the escape speed (VESC) in km/sec

VCIR = sqr(k/Ro)

VESC = sqr((2*k)/Ro) [Ref. 4:pp. 34-35]

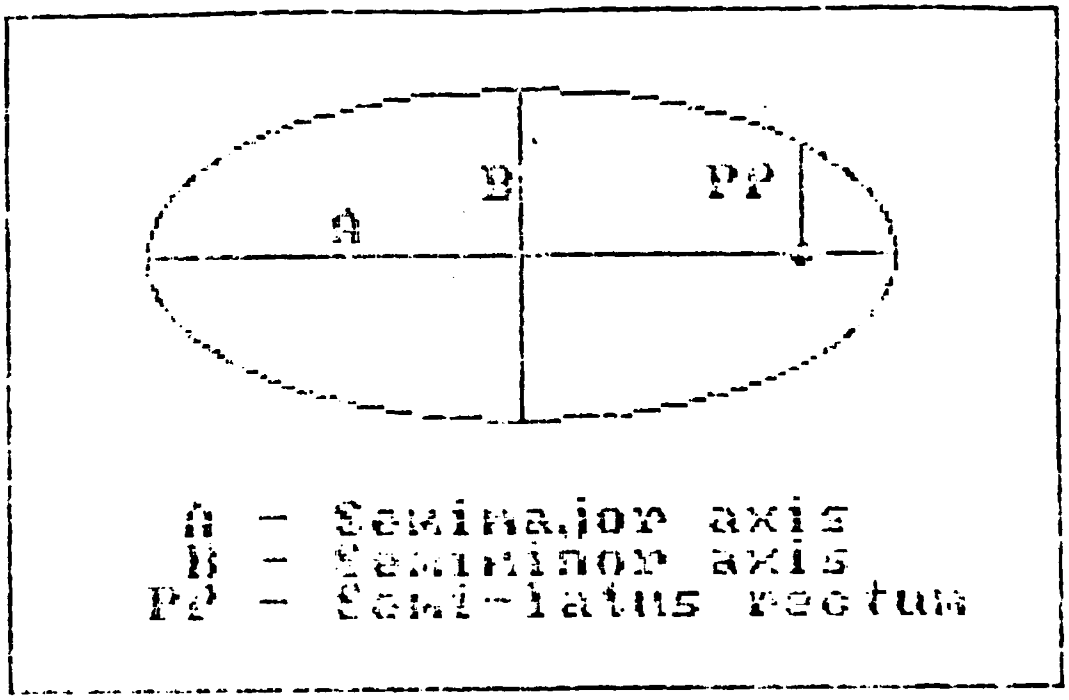


Figure 2. Geometrical dimensions of an orbit

If the orbit is an ellipse, the following orbital data is also calculated:

The semi-major axis(A), the semi-minor axis(B), and the semi-latus rectum (PP) in km

```
A = abs(Ro/(AAA-2))

B = sqr(A^2*(1-ECC^2))

PP= A*(1-ECC^2) [Ref. 7:pp. 97-98]
```

The radius at apogee (RAP), and the radius at perigee (RPE) in km

```
RAP = A*(1+ECC)

RPE = A*(1-ECC)
```

The velocities in km/sec at apogee (VAP), and at perigee (VPE) in km/sec

VAP = SAM/RAP

VPE = SAM/RPE [Ref. 4:p. 27]
The period (T), in minutes

T = SQR((4*pi^2*A^3)/k) [Ref. 4:p. 33]

If the orbit is circlular, the following data is calculated:

The eccentricity = 0.

The semi-major axis is the initial radius (Ro = A = B = PP).

The specific angular momentum (SAM), the specific mechanical energy (SME), and the period (T), using the same formulas as for an ellipse since the circle is special form of the ellipse.

If the orbit is parabolic, the following data is calculated:

The eccentricity (ECC) = 1.

The specific mechanical energy (SME) = 0.

The specific angular momentum (SAM), SAM > 0.

The altitude at perigee (Ro-Re).

The semi-latus rectum (PP).

The escape speed (VESC).

If the orbit is hyperbolic, the following is calculated:

The eccentricity (ECC). ECC > 1.

The altitude at perigee (Ro-Re).

The specific mechanical energy (SME).

The specific angular momentum (SAM).

The escape speed (VESC).

The hyperbolic excess speed (HES) in km/sec

HES = sqr(Vo^2-VESC^2) [Ref. 4:p. 40]

If the orbit intersects the earth's surface it is considered a ballistic trajectory and the following is calculated:

The maximum altitude above the surface in km

MAXALT = RAP-Re

The true anomaly at launch in degrees

THETAO = arccos((1/ECC)*(AAA*cos(Bo)^2-1))

The true anomaly at impact in degrees

THETAI = arccos((1/ECC)*((AAA*Ro)/Re)*cos(Bo)^2-1)

The range in km

RANGE = Re * (2*pi-THETAO-THETAI)

The eccentric anomaly at launch in degrees

EO = arccos((1/ECC)*(1-(Ro/A))

The eccentric anomaly at impact in degrees

EI = 2*pi-arccos((1/ECC)*(1-(Re/A))

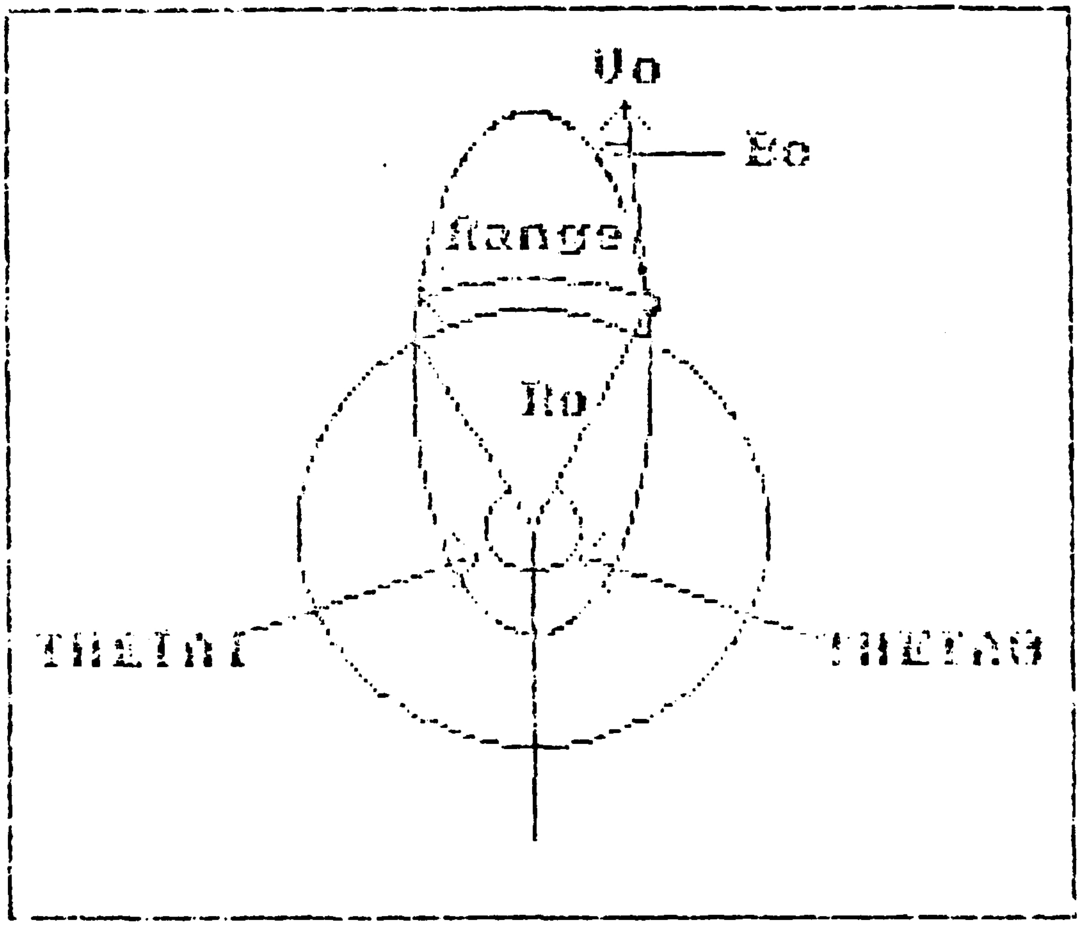


Figure 3. Ballistic trajectory

Time of flight in minutes

```
FLTTIME = sqr(A^3/k)*(EI-ECC*sin(EI)-EO+ECC*sin(EI))
```

Optimum launch angle in degrees for maximum range

```
P1 = 2/(1/(2*A-Re)+1/(2*A-Ro))
BOPT = arccos(sqr(k*P1)/(Ro*Vo))
```

If the launch angle is 90 degrees, the trajectory is considered a vertical launch and the following is calculated:

The maximum altitude in km

```
RMAX = (2*Ro)/(2-AAA)
MAXALT = RMAX - Re
```

The rise time in minutes

```
Q = arcsin(sqr(Ro/RMAX))
TRISE = .5*sqr(RMAX^3/(2*k))*(pi+sin(2*Q)-2*Q)
```

If AAA is greater than or equal to 2, then the satellite escapes.

After the orbital data is calculated, the program plots the orbit. A circle representing the earth is drawn using the CIRCLE command. The size of the circle is determined by the radius of the orbit (Ro). The screen coordinates (X,Y) are determined from the spherical coordinates (R, THETA). A loop is used to plot the orbit

```
FOR THETA = 0 to 2*pi STEP pi/30
```

The radius (R) in spherical coordinates is determined

If the orbit is

circular:

```
R = A
```

elliptical:

```
R = (A * (1-ECC^2))/(1+ECC*cos(THETA))
```

parabolic:

```
R = (2*RPE)/(1+cos(THETA))
```

hyperbolic:

```
R = (A * (ECC^2-1))/(1+ECC*cos(THETA))
```

vertical launch:

```
R = (2*Ro)/(2-AAA) where THETA = 0.
```

The screen coordinates, (X,Y), are calculated from the spherical coordinates

```
X = R * cos(THETA-THETAO)
Y = R * sin(THETA-THETAO)
```

These points are then plotted on the screen with the BASICA PSET command.

At the end of the loop, the command NEXT THETA causes the program to return to the start of the loop until the orbit is completed. The user is offered the opportunity to plot another orbit over the first. It must be an orbit with the same initial altitude because the size of circle representing the earth is determined by the altitude of the orbit. At the end of every orbit the user can elect to view the data associated with that orbit. If this option is chosen, then another orbit cannot be plotted over the previous orbit.

IV. PROGRAM ORBIT

The program INITIAL CONDITIONS plots a two-dimensional orbit in which size and shape of the orbit are determined. Program ORBIT takes this one step further and shows the orientation of the orbit in three-dimensions. Five independent "orbital elements" are traditionally used to describe the size, shape and orientation of an orbit. A sixth element, time, is used to pinpoint the position of the satellite along the orbit at a particular time. The six classical orbital elements are:

1. Semi-major axis (a) - a constant defining the sizeof the orbit. It is measured as half the distance from the perigee point to the apogee point.

2. Eccentricity (e) - a constant defining the shape of the orbit. It varies from zero for a circle, between zero and one for an ellipse, one for a parabola, and greater than one for a hyperbola.

3. Inclination (i) - the angle the orbital plane makes with the equatorial plane measured from the equator up in a counter-clockwise direction.

4. Longitude of the ascending node (Ω) - the point where the satellite crosses the equator in a northernly direction (ascending node) measured counter-clockwise from the prime meridian (0 degree point).

5. Argument of perigee (w) - the angle, measured along the orbital plane in the direction of the satellite's motion, from the ascending node to the perigee point.

6. Time of perigee passing (T) - the time the satellite is at the perigee point. The satellite position can be pinpointed by measuring the time elapsed from the time it passes the perigee point. [Ref. 4:p. 58]

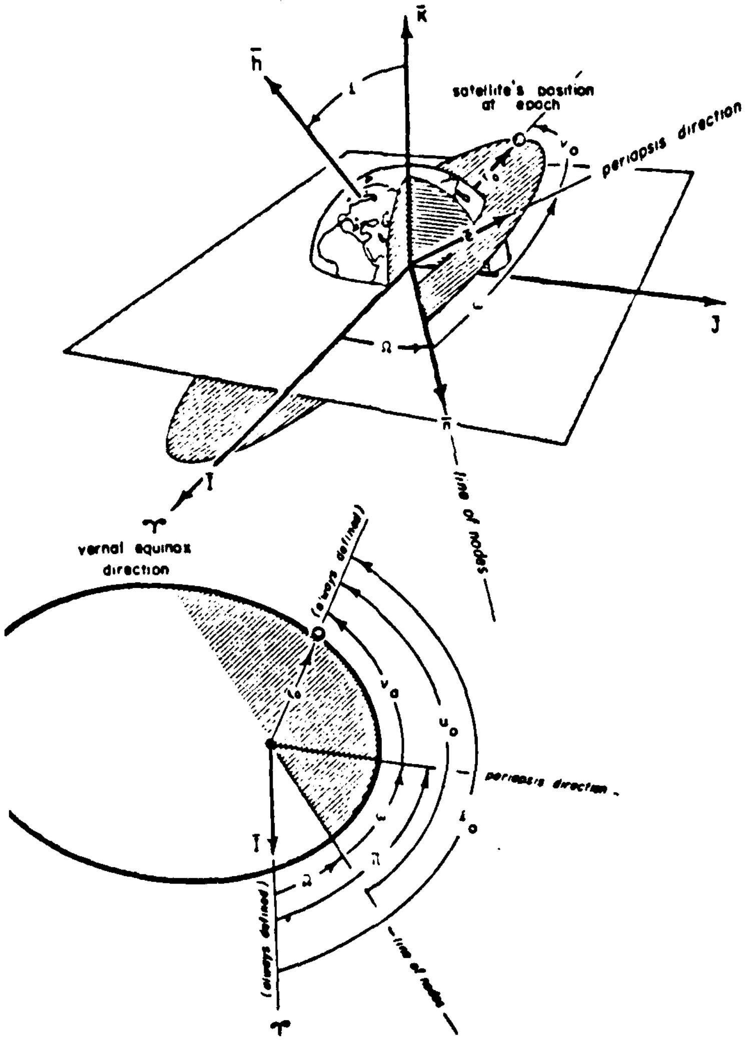


Figure 4. Orbital elements

The equations in this program came primarily from Methods of Orbit Determination by P. R. Escobal [Ref. 6] Any exceptions will be referenced.

ORBIT begins with a brief introduction followed by the user input of the first five classical orbital elements. If the eccentricity is zero, the program skips the input of the argument of perigee and longitude of the ascending node as they are not defined in the case of a circular orbit. In order to keep the program simple, it was decided to show one complete orbit from perigee point to perigee point. The sixth classical orbital element, time of perigee passage, would be measured from the perigee point.

Once the semi-major axis is entered, the program sizes the sphere representing the earth and draws it in the viewport.

The program converts the input from degrees to radians

```
TORAD = .01745329
X = X * TORAD
```

distances are converted to canonical units

```
Re = 6378.165 km
ER = A / Re
```

The next step in the program is to convert the coordinates of the satellite's position to the P-Q-R coordinate system

```
W = argument of perigee in radians
I = inclination in radians
O = ascending node in radians

PX = COS(W)*COS(O)-SIN(W)*SIN(O)*COS(I)
PY = COS(W)*SIN(O)+SIN(W)*COS(O)*COS(I)
PZ = SIN(W)*SIN(I)

QX = -SIN(W)*COS(O)-COS(W)*SIN(O)*COS(I)
QY = -SIN(W)*SIN(O)+COS(W)*COS(O)*COS(I)
QZ = COS(W)*SIN(I)
```

the mean motion is calculated

```
N = k / ER^1.5
```

the period is

```
P = TWOPI / N
```

The main program loop divides the orbit into time steps of the period divided by 50. The loop goes from the perigee point for one complete orbit back to the perigee point

```
FOR TTT = 0 TO P STEP 50
```

The first calculation within the loop is the solution to Kepler's equation. This is accomplished by a subroutine that uses another loop to find the eccentric anomaly (EN1). Lines 1620-1760 in the program listing contain the subroutine for the solution to Kepler's equation. This method came from Dr. Fuhs' class notes [Ref. 9].

The next step in the loop is to convert the coordinates to the XW-YW-RW coordinate system

```
XW = ER*(cos(EN1)-ECC)
YW = ER*sqr(1-ECC^2)*sin(EN1)
RW = ER*(1-ECC*cos(EN1))
```

the velocity is found

```
VEL = sqr(k * ((2/(RW*Re))-(1/A))) [Ref. 9]
```

the revolutions the satellite makes around the earth each day (N1) is

```
N1 = 16.997 * (1/ER)^1.5 [Ref. 5:p. 105]
```

The program then prints the velocity in km/sec, the revolutions per day (N), the orbital period in minutes, and the radius of the orbit in km, on the top portion of the screen.

If the orbit strikes the earth (suborbital), the program flashes a warning.

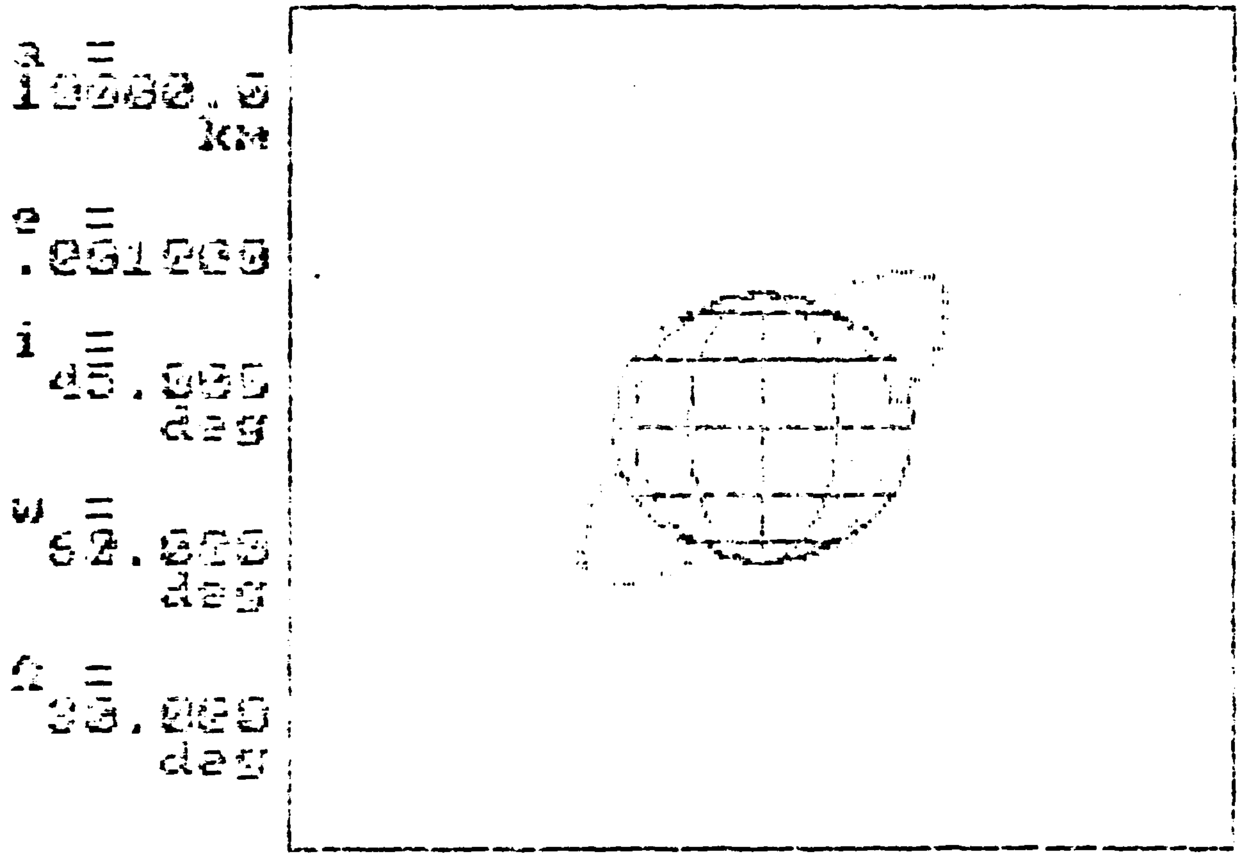


Figure 5. Example of Orbit Program

The next step in the loop converts the coordinates to the X-Y-Z coordinate system

```
X = XW*PX+YW*QX
Y = XW*PY+YW*QY
Z = XW*PZ+YW*QZ
```

Before plotting the coordinates to the screen, a hidden line removal algorithm is used to determine those coordinates hidden behind the globe. They are not plotted and the loop continues.

The CIRCLE command is used to plot the orbit around the globe. When the loop is finished the NEXT TTT command sends the program to the top of the loop until the orbit is complete. After the orbit is drawn, the option of drawing another orbit over the first is given. The user can also elect to return to the menu.

V. PROGRAM GROUND TRACK

This program takes for user input the same set of classical orbital elements as program ORBIT. It displays the ground trace of the satellite over the surface of the earth on a modified cartesian projection map of the world. The map was drawn pixel-by-pixel and the entire screen saved with the BSAVE command. At the start of the program it is loaded to a non-viewed screen with the BLOAD command. The non-viewed screen is then paged to the viewing screen.

This program is identical to ORBIT in converting the initial input to the X-Y-Z coordinate system. Instead of drawing a sphere, the program draws a map of the earth. The equations again are from Escobal [Ref. 6]. The exceptions are referenced.

To get the longitude from the X-Y-Z coordinates

```
O1 = ASCENDING NODE
ROTEAR = ROTATION OF THE EARTH = .004369 rad/min
ALPHA = arctan(Y/X)
LONGITUDE = ALP  -(ROTEAR*TTT)+O1
```

The most difficult programming problem of this project was to keep the longitude in the correct quadrant after taking the arctangent of Y divided by X. The solution to the problem is found in the subroutine in lines 1330-1410 in the program listing.

The latitude is found from the X-Y-Z coordinate system

```
R = sqr(X^2+Y^2+Z^2)
LAT = arcsin(Z / R)
```

The program changes the latitude and longitude from radians to degrees and there are checks in the program to keep the longitude between 0 and 360 degrees and the latitude between +90 and -90 degrees.

The latitude and longitude are then scaled to the screen and plotted from the perigee point using the CIRCLE command. GROUND TRACK has the option of drawing more than one orbit in order to see what the intertrace spacing looks like. The intertrace spacing is the distance on the earth between the first and subsequent *ground* tracks.

After the ground track for a particular satellite is finished, the user can enter the orbital elements to plot the ground track of a different satellite or return to the menu.

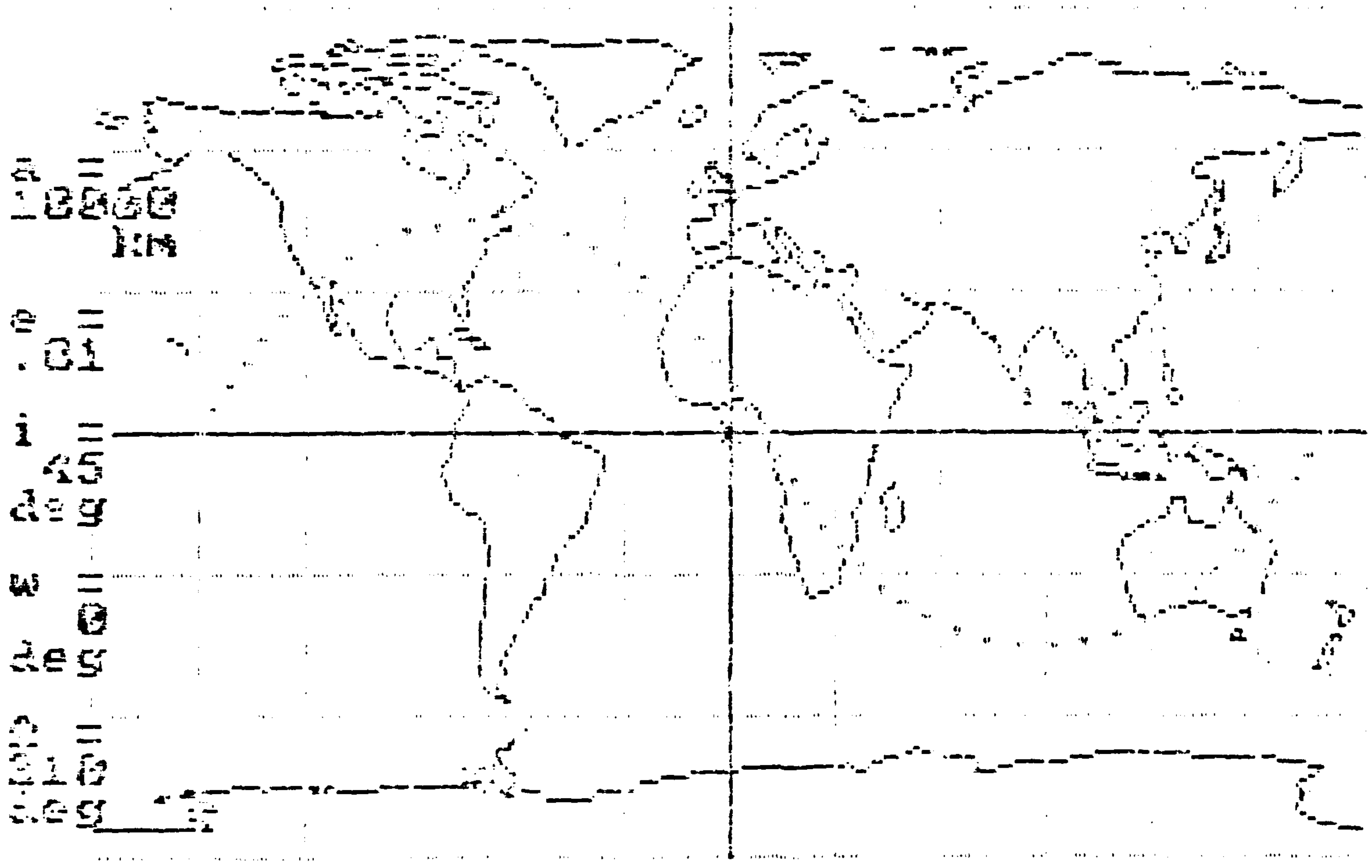


Figure 6. Example of Ground Track Program

VI. PROGRAM GEOSYNC

A geosynchronous orbit is a special case of the circular orbit. The altitude must be constant at about 19,300 nm., 5.6 earth radii, or 35,750 km. The period is 23 hours and 56 minutes. The only variables are the inclination and the longitude of the equator crossing. If a geosynchronous satellite has an inclination of zero degrees, then it will appear to hover motionless over a point on the equator. It is then called a geostationary satellite. The general pattern of the ground track of a geosynchronous satellite is a figure eight about the equator.

Program GROUND TRACK can plot a geosynchronous orbit, however it is very slow in doing so. Dr. Allen E. Fuhs, Distinguished Professor of Aeronautics at the Naval Postgraduate School, presented an alternate set of equations to plot the ground track of a geosynchronous orbit [Ref. 9]. The computation time is much shorter, therefore the computer can plot it much quicker.

GEOSYNC uses the same map as GROUND TRACK. After the user inputs the inclination and longitudinal crossing point, the program calculates the rotation rate of the earth

```
We = (2*pi) / 24 radians per hour
```

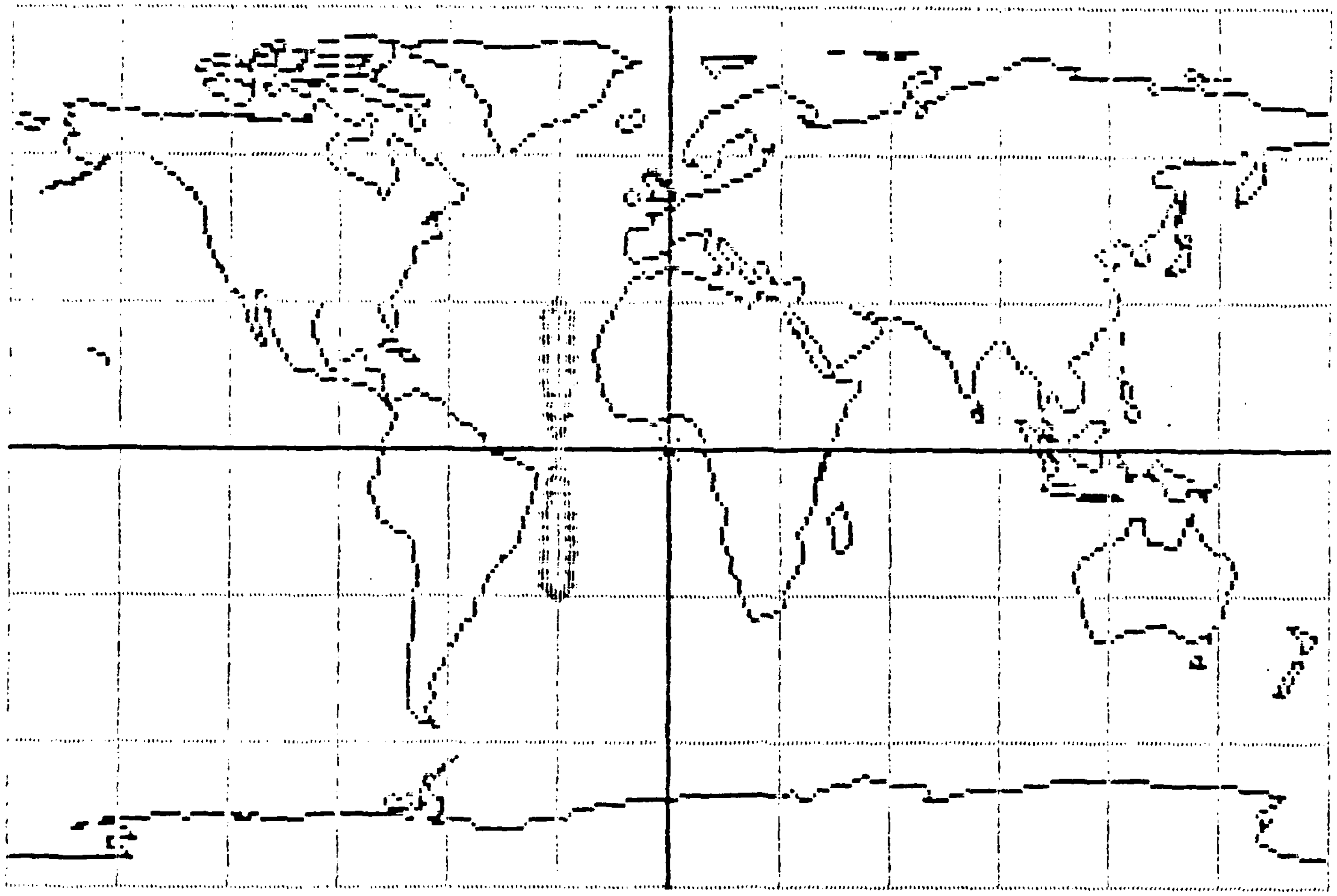


Figure 7. Example of Geosync Program

the rotation rate of the satellite equals the rotation rate of the earth

```
  Wo = We rad/hr
```

The time loop is in hours and goes from 0 to 18 with steps of .25 hours

```
  For T = 0 TO 18 STEP .25
```

The latitude is calculated

```
  LAT = ARCSIN(SIN(I)*SIN(Wo*T))
```

and the longitude is

```
  LONG = ATN(COS(I)*TAN(Wo*T))-We*T
```

The latitude and longitude are then converted back to degrees, scaled to the screen, and plotted to the map of the world with the CIRCLE command.

An additional factor that needed to be dealt with in this program was the adjustment necessary when the orbit was located in the 4th quadrant (X and Y are both negative). In this case another loop was necessary. The only difference between the main loop and the 4th quadrant loop was the addition of 2 times pi to the longitude.

VII. RECOMMENDATIONS FOR FURTHER DEVELOPMENT

These programs are meant to be used by beginning space science students. They were kept at an introductory level. As such, they should be considered as an initial attempt and improved upon. The next step from a simple Keplerian orbit would be to add the various perturbations that act on a real artificial satellite. These include the effect of other attracting bodies (the moon), atmospheric drag and lift, the oblateness of the earth, magnetic torque, and solar radiation pressure [Ref. 4:p. 386] Perturbations must be considered to achieve reasonable accuracy as far as predicting satellite position over a long period of time.

Another idea for further development would be to write programs for basic orbital maneuvers or rendezvous problems. Lunar, interplanetary, and missile trajectories could also be programmed. The equations for these programs are found in several texts on astrodynamics [Ref. 4, 6, 7, 8].

The Air Force Academy and The Naval Academy have orbital programs that run on mini-computers that support graphics. A good project would be to develop a similar capability at The Naval Postgraduate School. The graphics capabilities of the larger computers are much greater than those of most microcomputers. Two new microcomputers that have tremendous graphics capabilities are the Amiga and the

Atari ST-1040. Orbital graphics programs written for these two machines would be able to achieve better graphics than on most older microcomputers.

A final recommendation is that the GROUND TRACK program could be modified to show ground coverage, number of satellites in a constellation, and other elements necessary for planning a real-world artificial satellite application.

APPENDIX A

INTRO PROGRAM LISTING

```
10 ' INTRO
20 '
30 'This is the introduction and menu program
40 'that controls the other programs.
50 '
60 '
70 'set up and draw the box around the screen
80 '
90 CLS:SCREEN 1,0:KEY OFF
100 COLOR 9
110 '
120 'title page
130 '
140 GOSUB 870
150 '
160 'description of programs
170 '
180 GOSUB 610
190 '
200 '
210 'draw the box around menu
220 LINE(0,0)-(319,199),1,B
230 LINE(10,10)-(309,189),1,B
240 PAINT(5,5),1
250 '
260 'print the choices on the screen
270 '
280 LOCATE 5,17
290 PRINT"PROGRAMS:
300 LOCATE 9,12
310 PRINT"INITIAL CONDITIONS
320 LOCATE 11,12
330 PRINT"ORBIT
340 LOCATE 13,12
350 PRINT"GROUND TRACK
360 LOCATE 15,12
370 PRINT"GEOSYNC
380 LOCATE 17,12
390 LOCATE 20,10
400 PRINT"TO RUN THE PROGRAM
410 LOCATE 21,10
420 PRINT"TYPE FIRST TWO LETTERS
430 '
440 '
450 'accept choice from keyboard
460 '
```

```
470 X$=INPUT$(2)
480 '
490 'load the chosen program
500 '
510 IF X$="ic" THEN LOAD"INCL",R
520 IF X$="ec" THEN LOAD"ECC",R
530 IF X$="le" THEN LOAD"LEM",R
540 IF X$="in" THEN LOAD"ORBIT",R
550 IF X$="or" THEN LOAD"3DORB1",R
560 IF X$="gr" THEN LOAD"ELLIPORB",R
570 IF X$="ge" THEN LOAD"GEOSYNC",R ELSE 470
580 '
590 END
600 '
610 'introduction
620 '
630 LOCATE 3,1
640 PRINT"   These orbital programs demonstrate
650 PRINT"several principals of orbital flight.
660 PRINT
670 PRINT"INITIAL CONDITIONS is a graphical
680 PRINT"representation of the two-body
690 PRINT"problem.
700 PRINT
710 PRINT"ORBIT accepts the classical orbital
720 PRINT"elements and draws the orbit around
730 PRINT"a sphere representing the earth.
740 PRINT
750 PRINT"GROUND TRACK charts the ground trace
760 PRINT"of the orbit over a map of the earth
770 PRINT
780 PRINT"GEOSYNC traces the ground track of a
790 PRINT"geosynchronous orbit over a map of the
800 PRINT"earth.
810 LOCATE 24,1
820 INPUT"press enter to continue....",KAL
830 CLS
840 RETURN
850 STOP
860 '
870 'title page
880 '
890 LINE(5,5)-(315,150),2,B
900 LOCATE 4,4:PRINT"SATELLITE ORBIT PROGRAMS UTILIZING
910 LOCATE 6,6:PRINT"THE GRAPHICS CAPABILITES OF THE
920 LOCATE 8,14:PRINT"MICROCOMPUTER
930 LOCATE 11,19:PRINT"by
940 LOCATE 14,10:PRINT"Kim Alldredge Langdorf
950 LOCATE 16,15:PRINT"June 1986
960 LOCATE 22,7
970 INPUT"press enter to continue....",KAL
980 CLS
```

```
990 RETURN
999 STOP
```

APPENDIX B

INITIAL CONDITIONS PROGRAM LISTING

```
10 'initial conditions
20 '
30 'set up the screen
40 '
50 CLS:KEY OFF:SCREEN 1,0
60 COLOR 9
70 IF KAL<>0 THEN GOTO 110
80 GOSUB 4810
90 '
100 'constants
110 '
120 PI=3.1415927#
130 K=398603.2
140 RE=6378.165
150 '
160 'counter to prevent redrawing of sphere
170 '
180 IF KAL<>0 THEN GOTO 260
190 '
200 'draw sphere
210 '
220 GOSUB 4450
230 '
240 'set viewport on screen
250 '
260 VIEW(95,30)-(310,190),,3
270 CLS:KAL=0
280 '
290 'accept user input and echo on screen
300 '
310 LOCATE 4,1:PRINT"          ":PRINT"          ":LOCATE
7,1:PRINT"          ":PRINT"          ":LOCATE 10,1:PRINT"
":PRINT"          ":LOCATE 18,1:PRINT"          ":PRINT"
":LOCATE 21,1:PRINT"          ":PRINT"          "
320 RO=0:VO=0:BO=0:RAP=0
330 LOCATE 1,1
340 INPUT"initial altitude in km = ",ALT
350 RO=ALT+RE
360 LOCATE 1,5:PRINT"                              "
370 LOCATE 4,1:PRINT"Ro (km)  =
380 LOCATE 5,1:PRINT USING"#######.##";RO
390 LOCATE 1,1
400 INPUT"initial flight path angle in deg = ",BETADEG

410 LOCATE2,1:PRINT"                                    ":
LOCATE 1,1
420 IF BETADEG>90 GOTO 4120
```

```
430 LOCATE 1,1:PRINT"
"
440  LOCATE  7,1:PRINT"Bo  (deg)  ="
450  LOCATE  8,1:PRINT  USING"#######.##";BETADEG
460  KAL=KAL+1
470  BETA=BETADEG*(PI/180)
480  LOCATE  1,3
490  PRINT"would  you  like  to  input  initial
500  LOCATE  2,3
510  PRINT"speed,  Vo,  or  desired  apogee  ,Ap?
520  W$=INPUT$(1)
530  IF  W$="v"  THEN  GOTO  550
540  IF  W$="a"  THEN  GOTO  4320  ELSE  520
5 5 0                    L O C A T E                   1 , 1 : P R I N T "
":PRINT"                                              ":LOCATE 1,1
560  INPUT"initial  speed  in  km/s  =  ",VO
570  IF  VO=0  THEN  BEEP:GOTO  550
580  LOCATE  10,1:PRINT"Vo  (km/s)=":LOCATE  11,1:PRINT
USING"#######.##";VO
590  LOCATE  1,1
600  PRINT"                                            "
610  '
620  'calculations  &  type  of  orbit  determination
630  '
640  VCIR=SQR(K/RO)
650  VESC=SQR((2*K)/RO)
660  SAM=VO*RO*COS(BETA)
670  ENERGY=(VO^2/2)-(K/RO)
680  ALPHA=(RO*VO^2)/K
690  E=SQR((ALPHA-1)^2*(COS(BETA))^2+(SIN(BETA))^2)
700  IF  ALPHA=2  THEN  SSS=3:GOTO  3160
710  A=ABS(RO/(ALPHA-2))
720  PP=A*(1-E^2)
730  IF  ALPHA  >  2  THEN  SSS=4:GOTO  3090
740  IF  ABS(E-0)<.0001  THEN  SSS=2:A=RO:R$="circular  orbit":
GOTO  790
750  RPE=A*(1-E)
760  IF  BETADEG=90  THEN  SSS=6:GOTO  3240
770  SSS=1:RAP=A*(1+E)
780  R$="elliptical  orbit"
790  T=SQR((4*PI^2*A^3)/K)
800  '
810  'to  graphics  routine
820  '
830  GOTO  3650
840  STOP
850  '
860  'label  strings
870  '
880  CLS
890  A$="circular  orbit"
900  B$="elliptical  orbit"
```

```
910 C$="parabolic orbit"
920 D$="hyperbolic orbit"
930 E$="circular orbit"
940 F$="ballistic flight"
950 G$="vertical launch"
960 H$="semi-major axis    ="
970 I$="semi-latus rectum ="
980 J$="eccentricity       ="
990 K$="alt at apogee      ="
1000 L$="alt at perigee     ="
1010 M$="vel at apogee      ="
1020 N$="vel at perigee     ="
1030 O$="period             ="
1040 P$="sp mech energy     ="
1050 Q$="sp ang momentum    ="
1060 S$="escape velocity    ="
1070 T$="circular velocity ="
1080 U$="max alt above sfc ="
1090 V$="range              ="
1100 W$="flight time        ="
1110 XX$="vertical max alt ="
1120 YY$="rise time (min)   ="
1130 UU$="alt above sfc     ="
1140 HH$="semi-minor axis    ="
1150 '
1160 'display data
1170 '
1180 IF SSS=1 GOTO 1250
1190 IF SSS=2 GOTO 1710
1200 IF SSS=3 GOTO 1960
1210 IF SSS=4 GOTO 2180
1220 IF SSS=5 GOTO 2440
1230 IF SSS=6 GOTO 2640
1240 '
1250 'display elliptical orbit data
1260 '
1270 B=SQR(A^2*(1-E^2))
1280 LOCATE 4,10:PRINT R$
1290 LOCATE 8,1:PRINT HH$
1300 LOCATE 8,20:PRINT USING"#######.##";B
1310 LOCATE 8,32:PRINT"km"
1320 LOCATE 7,1:PRINT H$
1330 LOCATE 7,20:PRINT USING"#######.##";A
1340 LOCATE 7,32:PRINT"km"
1350 LOCATE 9,1:PRINT I$
1360 LOCATE 9,20:PRINT USING"#######.##";PP
1370 LOCATE 9,32:PRINT"km"
1380 LOCATE 10,1:PRINT J$
1390 LOCATE 10,20:PRINT USING"#####.#####";E
1400 LOCATE 11,1:PRINT K$
1410 LOCATE 11,20:PRINT USING"#######.##";RAP-RE
1420 LOCATE 11,32:PRINT"km"
```

```
1430 LOCATE 12,1:PRINT L$
1440 LOCATE 12,20:PRINT USING"########.##";RPE-RE
1450 LOCATE 12,32:PRINT"km"
1460 LOCATE 13,1:PRINT M$
1470 VAP=SAM/RAP
1480 LOCATE 13,20:PRINT USING"#####.#####";VAP
1490 LOCATE 13,32:PRINT"km/s"
1500 VPE=SAM/RPE
1510 LOCATE 14,1:PRINT N$
1520 LOCATE 14,20:PRINT USING"#####.#####";VPE
1530 LOCATE 14,32:PRINT"km/s"
1540 LOCATE 15,1:PRINT O$
1550 LOCATE 15,20:PRINT USING"#####.#####";T/60
1560 LOCATE 15,32:PRINT"min"
1570 LOCATE 16,1:PRINT P$
1580 LOCATE 16,20:PRINT USING"#####.#####";ENERGY
1590 LOCATE 16,32:PRINT"km"CHR$(253)"/s"CHR$(253)
1600 LOCATE 17,1:PRINT Q$
1610 LOCATE 17,20:PRINT USING"########.##";SAM
1620 LOCATE 17,32:PRINT"km"CHR$(253)"/s"
1630 LOCATE 18,1:PRINT S$
1640 LOCATE 18,20:PRINT USING"#####.#####";VESC
1650 LOCATE 18,32:PRINT"km/s
1660 LOCATE 19,1:PRINT T$
1670 LOCATE 19,20:PRINT USING"#####.#####";VCIR
1680 LOCATE 19,32:PRINT"km/s
1690 GOTO 2740
1700 '
1710 'display circular orbit data
1720 '
1730 LOCATE 11,10:PRINT A$
1740 LOCATE 13,1:PRINT J$
1750 LOCATE 13,20:PRINT USING"####.####";E
1760 LOCATE 14,1:PRINT UU$
1770 LOCATE 14,20:PRINT USING"######.##";RO-RE
1780 LOCATE 14,32:PRINT"km
1790 LOCATE 15,1:PRINT O$
1800 LOCATE 15,20:PRINT USING"#####.#####";
SQR((4*PI^2*A^3)/K)/60
1810 LOCATE 15,32:PRINT"min"
1820 LOCATE 16,1:PRINT P$
1830 LOCATE 16,20:PRINT USING"#####.#####";-K/(2*A)
1840 LOCATE 16,32:PRINT"km"CHR$(253)"/s"CHR$(253)
1850 LOCATE 17,1:PRINT Q$
1860 LOCATE 17,20:PRINT USING"########.##";RO*SQR(K/RO)
1870 LOCATE 17,32:PRINT"km"CHR$(253)"/s"
1880 LOCATE 18,1:PRINT S$
1890 LOCATE 18,20:PRINT USING"#####.#####";SQR((2*K)/RO)
1900 LOCATE 18,32:PRINT"km/s
1910 LOCATE 19,1:PRINT T$
1920 LOCATE 19,20:PRINT USING"#####.#####";SQR(K/RO)
1930 LOCATE 19,32:PRINT"km/s
```

```
1940 GOTO 2740
1950 '
1960 'display parabolic orbit data
1970 '
1980 LOCATE 10,10:PRINT C$
1990 LOCATE 12,1:PRINT J$
2000 LOCATE 12,20:PRINT USING"#####.#####";E
2010 LOCATE 13,1:PRINT L$
2020 LOCATE 13,20:PRINT USING"#######.##";ABS(RPE-RE)
2030 LOCATE 13,32:PRINT"km
2040 LOCATE 14,1:PRINT P$
2050 LOCATE 14,20:PRINT USING"#######.##";ENERGY
2060 LOCATE 14,32:PRINT"km"CHR$(253)"/s"CHR$(253)
2070 LOCATE 16,1:PRINT S$
2080 LOCATE 15,1:PRINT Q$
2090 LOCATE 15,20:PRINT USING"#######.##";SAM
2100 LOCATE 15,32:PRINT"km"CHR$(253)"/s"
2110 LOCATE 16,20:PRINT USING"#######.##";VESC
2120 LOCATE 16,32:PRINT"km
2130 LOCATE 17,1:PRINT I$
2140 LOCATE 17,20:PRINT USING"#######.##";PP
2150 LOCATE 17,32:PRINT"km
2160 GOTO 2740
2170 '
2180 'display hyperbolic orbital data
2190 '
2200 LOCATE 10,10:PRINT D$
2210 LOCATE 12,1:PRINT J$
2220 LOCATE 12,20:PRINT USING"#####.#####";E
2230 LOCATE 13,1:PRINT L$
2240 LOCATE 13,20:PRINT USING"#######.##";ABS(RPE-RE)
2250 LOCATE 13,32:PRINT"km
2260 LOCATE 14,1:PRINT P$
2270 LOCATE 14,20:PRINT USING"#######.##";ENERGY
2280 LOCATE 14,32:PRINT"km"CHR$(253)"/s"CHR$(253)
2290 LOCATE 15,32:PRINT"km"CHR$(253)"/s"
2300 LOCATE 15,1:PRINT Q$
2310 LOCATE 15,20:PRINT USING"#######.##";SAM
2320 LOCATE 16,1:PRINT S$
2330 LOCATE 16,20:PRINT USING"#######.##";VESC
2340 LOCATE 16,32:PRINT"km
2350 LOCATE 17,1:PRINT"hyperbolic exc sp =
2360 HES=SQR(VO^2-VESC^2)
2370 LOCATE 17,20:PRINT USING"#######.##";HES
2380 LOCATE 17,32:PRINT"km
2390 LOCATE 18,1:PRINT I$
2400 LOCATE 18,20:PRINT USING"#######.##";ABS(PP)
2410 LOCATE 18,32:PRINT"km
2420 GOTO 2740
2430 '
2440 'display suborbital data
2450 '
```

```
2460 LOCATE 10,10:PRINT F$
2470 LOCATE 12,1:PRINT U$
2480 LOCATE 12,20:PRINT USING"#######.##";RAP-RE
2490 LOCATE 12,32:PRINT"km
2500 LOCATE 13,1:PRINT V$
2510 LOCATE 13,20:PRINT USING"#######.##";RANGE
2520 LOCATE 13,32:PRINT"km
2530 LOCATE 14,1:PRINT W$
2540 LOCATE 14,20:PRINT USING"#####.#####";FLTTIME/60
2550 LOCATE 14,32:PRINT"min
2560 LOCATE 15,1:PRINT"optimum Bo for Vo =
2570 LOCATE 15,20:PRINT USING"#####.#####";BOPT*(180/PI)
2580 LOCATE 15,32:PRINT"deg
2590 GOTO 2740
2600 STOP
2610 '
2620 'display vertical launch data
2630 '
2640 LOCATE 10,10:PRINT R$
2650 LOCATE 12,1:PRINT XX$
2660 LOCATE 12,20:PRINT USING"#######.##";RMAX-RE
2670 LOCATE 12,32:PRINT"km
2680 LOCATE 13,1:PRINT YY$
2690 LOCATE 13,20:PRINT USING"#######.##";TRISE
2700 LOCATE 13,32:PRINT"min
2710 '
2720 'choice of another satellite
2730 '
2740  LOCATE 1,1:PRINT"
":LOCATE 1,1
2750 PRINT"another satellite?
2760 X$=INPUT$(1)
2770 IF X$="n" THEN LOAD"pick",R
2780 IF X$="y" THEN GOTO 50 ELSE 2740
2790 STOP
2800 '
2810 'subroutine to calculate true anomaly
2820 '
2830 X=(RO*VO^2)/K
2840 Y=(ALPHA*SIN(BETA)*COS(BETA))/(X*COS(BETA)^2-1)
2850 THETAO=ATN(Y)
2860 RETURN
2870 '
2880 'choice of viewing orbital data, another
2890 'satellite, or return to menu
2900 '
2910 LOCATE 1,1
2920 PRINT"                                    "
2930 LOCATE 2,1
2940 PRINT"                                    "
2950 LOCATE 1,1
2960 PRINT"again from same altitude?"
```

```
2970 X$=INPUT$(1)
2980 IF X$="y" THEN LOCATE 1,1:PRINT"
":GOTO 390
2990 IF X$="n" THEN 3000 ELSE 2950
3000 LOCATE 1,1
3010 PRINT"would you like to see orbital data?
3020 Y$=INPUT$(1)
3030 IF Y$="y" THEN SCREEN 0,0:COLOR 9:GOTO 880
3040 IF Y$="n" THEN 2740 ELSE 3000
3050 STOP
3060 '
3070 'calculations for hyperbolic orbit
3080 '
3090 R$="hyperbolic orbit"
3100 RPE=A*(E-1)
3110 GOTO 3650
3120 STOP
3130 '
3140 'calculations for parabolic orbit
3150 '
3160 R$="parabolic orbit"
3170 RPE=.5*RO*ALPHA*(COS(BETA))^2
3180 PP=RPE*(1+E)
3190 GOTO 3650
3200 STOP
3210 '
3220 'calculations for vertical launch
3230 '
3240 R$="vertical launch"
3250 IF ALPHA >= 2 THEN LOCATE 2,15:PRINT"vert escapes":GOTO
3300
3260 RMAX=(2*RO)/(2-ALPHA)
3270 F=SQR(RO/RMAX)
3280 Q=ATN(F/SQR(-F*F+1))
3290 TRISE=.5*SQR(RMAX^3/(2*K))*(PI+SIN(2*Q)-2*Q)
3300 GOTO 3650
3310 STOP
3320 '
3330 'calculations for ballistic trajectory
3340 '
3350 R$="suborbital flight
3360 SSS=5
3370 RAP=A*(1+E)
3380 X=(1/E)*(ALPHA*COS(BETA)^2-1)
3390 IF X=1 THEN X=.999999
3400 IF X=-1 THEN X=-.999999
3410 THETA1=-ATN(X/SQR(-X*X+1))+1.5708
3420 Y=(1/E)*((ALPHA*RO/RE*COS(BETA)^2)-1)
3430 IF Y=1 THEN Y=.999999
3440 IF Y=-1 THEN Y=-.999999
3450 THETAE=-ATN(Y/SQR(-Y*Y+1))+1.5708
3460 RANGE=(2 * PI - THETA1-THETAE) * RE
```

```
3470 Z=(1/E)*(1-(RO/A))
3480 IF Z=-1 THEN Z=-.999999
3490 IF Z=1 THEN Z=.999999
3500 EO=-ATN(Z/SQR(ABS(-Z*Z+1)))+1.5708
3510 Q=(1/E)*(1-(RE/A))
3520 IF Q=-1 THEN Q=-.999999
3530 IF Q=1 THEN Q=.999999
3540 P=-ATN(Q/SQR(ABS(-Q*Q+1)))+1.5708
3550 EE=2*PI-P
3560 FLTTIME=SQR(A^3/K)*(EE-E*SIN(EE)-EO+E*SIN(EO))
3570 P1=2/((1/(2*A-RE))+(1/(2*A-RO)))
3580 QQ=SQR(K*P1)/(RO*VO)
3590 IF QQ=-1 THEN QQ=-.999999
3600 IF QQ=1 THEN QQ=.999999
3610 BOPT=-ATN(QQ/SQR(ABS(-QQ*QQ+1)))+1.5708
3620 GOTO 2880
3630 STOP
3640 '
3650 'graphics subroutine
3660 '
3670 PPP=2*PI+.1
3680 LOCATE 1,15
3690 PRINT R$
3700 IF KAL <> 1 THEN GOTO 3720
3710 GOSUB 4200
3720 GOSUB 2800
3730 IF SSS=6 THEN PPP=PI/2
3740 IF BETA>THETAO THEN THETAO=THETAO+PI
3750 IF BETADEG=90 THEN OOO=PI/2:PPP=PI*1.5:GOTO 3800
3760 IF BETADEG<>0 AND ALT<>0 GOTO 3800
3770 IF BETADEG<>0 THEN THETAO=THETAO+PI
3780 IF RPE<RE-.1 THEN THETAO=THETAO+PI
3790 OOO=0
3800 FOR THETA=OOO TO PPP STEP PI/30
3810 KL=KL+1
3820 IF SSS=1 THEN R=(A*(1-E^2))/(1+E*COS(THETA))
3830 IF SSS=2 THEN R=A
3840 IF SSS=3 THEN R=(2*RPE)/(1+COS(THETA))
3850 IF SSS=4 THEN R=(A*(E^2-1))/(1+E*COS(THETA))
3860 IF SSS=6 THEN R=RMAX/RE:Y=0:GOTO 3910
3870 IF SSS=5 THEN R=A*(1-E^2)/(1+E*COS(THETA))
3880 R=R/RE
3890 Y=R*SIN(THETA-THETAO)
3900 Y=Y*(215/160)*(5/6)
3910 X=R*COS(THETA-THETAO)
3920 IF SSS=3 AND Y>RD THEN GOTO 4090
3930 IF SSS=4 AND X<-RD THEN GOTO 4090
3940 IF SSS=3 AND X<-RD THEN GOTO 4090
3950 IF SSS=4 AND Y>RD THEN GOTO 4090
3960 'IF POINT(X,Y)=1 THEN PRESET(X,Y),1:GOTO 10070
3970 PRESET(X,Y),1
3980 LINE-(X,Y),KAL+1,,&HAAAA
```

```
3990 IF SSS=4 OR SSS=3 GOTO 4080
4000 PSET(X,Y),KAL+1
4010 VEL=SQR(K*(2/(R*RE)-1/A))
4020 LOCATE 18,1:PRINT"Vo (km/s)=
4030 LOCATE 19,1:PRINT USING"######.###";VEL
4040 LOCATE 21,1:PRINT"alt (km)=
4050 LOCATE 22,1:PRINT USING"######.###";(R*RE)-RE
4060 IF KL=1 GOTO 4080
4070 IF R<.99 THEN SSS=5:LOCATE 2,15:PRINT"   suborbital
4080 NEXT THETA
4090 IF SSS=5 GOTO 3330
4100 GOSUB 2880
4110 '
4120 'limit Bo from 0 to 90 degrees
4130 '
4140 BEEP
4150 LOCATE 2,1
4160 PRINT"choose an angle between 0 and 90
4170 GOTO 390
4180 STOP
4190 '
4200 'draw circle representing earth
4210 '
4220 XO=0:YO=0:RD=(A/RE)*2
4230 IF SSS=3 THEN RD=(PP/RE)*2
4240 VIEW(95,30)-(310,190),,3
4250 WINDOW(XO-RD,YO-RD)-(XO+RD,YO+RD)
4260 CIRCLE(0,0),.98,1
4270 PAINT(0,0),1
4280 CIRCLE(RO/RE,0),RD/75,KAL+1
4290 PAINT(RO/RE,0),KAL+1
4300 RETURN
4310 STOP
4320 '
4330 'calculate Vo from apogee height
4340 '
4350LOCATE1,1:PRINT"
":PRINT"                                        ":LOCATE 1,1
4360 INPUT "apogee altitude (km) = ",RAP
4370 RAP=RAP + RE
4380 X=RO/RAP
4390 IF RO = RAP GOTO 2550
4400 ALPHA=((2-X-X^2*COS(BETA)^2)-SQR((2-X-X^2
*COS(BETA)^2)^2-4*(1-X)*(1-X^2*COS(BETA)^2)))
/(1-X^2*COS(BETA)^2)
4410 VO=SQR((ALPHA*K)/RO)
4420 GOTO 580
4430 STOP
4440 '
4450 'draw demonstration picture
4460 '
4470 CLS
```

```
4480 CIRCLE(160,105),55,1
4490 PAINT(160,105),1
4500 PSET(160,105),3
4510 DRAW"r80;u40;
4520 PSET(240,105),2
4530 DRAW"e40;l5;r5;d5;
4540 LOCATE 7,28:PRINT"local
4550 LOCATE 8,27:PRINT"horizon
4560 LOCATE 15,25
4570 PRINT"Ro
4580 LOCATE 8,36:PRINT"Vo
4590 CIRCLE(240,105),2,2
4600 PAINT(241,104),2
4610 PAINT(239,106),2
4620 CIRCLE(240,105),20,3,.27*PI,.5*PI
4630 PSET(248,93)
4640 DRAW"r5;u5
4650 LOCATE 11,32
4660 PRINT"Bo
4670 LOCATE 1,1
4680 PRINT"Alt = height above surface of earth
4690 PRINT"Re = radius of earth = 6378.165 km
4700 PRINT"Ro = Alt + Re = initial radius
4710 LOCATE 13,28:PRINT"Alt
4720 LOCATE 13,24:PRINT"Re
4730 LOCATE 4,1
4740 PRINT"Bo = initial flight path angle
4750 PRINT"Vo = initial speed
4760 LOCATE 23,1
4770 INPUT"press enter to continue...",LLL
4780 CLS
4790 RETURN
4800 '
4810 ' introduction
4820 '
4830 LOCATE 2,10
4840 PRINT"INITIAL CONDITIONS
4850 LOCATE 4,1
4860 PRINT"This program demonstrates the two-body
4870 PRINT"problem.  Enter the initial values for
4880 PRINT"altitude, flight path angle, and either
4890 PRINT"speed or desired apogee height.
4900 PRINT
4910 PRINT"The program then calculates various
4920 PRINT"characteristics of the resulting orbit
4930 PRINT"and plots the orbit around the earth.
4940 PRINT"You then have the option of viewing
4950 PRINT"the orbital data or plotting another
4960 PRINT"orbit on top of the first.
4970 PRINT
4980 PRINT"If the program isn't working properly,
4990 PRINT"press the Ctrl key and the Break key
```

```
5000 PRINT"together and then press the F2 key
5010 PRINT"to restart.  If you would like to make
5020 PRINT"a copy of the screen on the printer
5030 PRINT"press the shift key and the PrtSc key
5040 PRINT"at the same time.
5050 LOCATE 24,1
5060  INPUT"press  enter  to  continue...",LLL
5070 CLS
5080 RETURN
5090 STOP
```

APPENDIX C

ORBIT PROGRAM LISTING

```
10 'orbit
20 '
30 'set up screen
40 '
50 KEY OFF:CLS:SCREEN 1,0
60 COLOR 9
70 '
80 'introduction
90 '
100 GOSUB 2300
110 '
120 'set viewport
130 '
140 VIEW(60,20)-(260,180),,3
150 '
160 'constants
170 '
180 PI=3.141593
190 TWOPI=2*PI
200 TORAD=1.745329E-02
210 TODEG=57.2957795#
220 K=7.436575E-02:DTHETA=.004369
230 '
240 'routine to space input
250 '
260 GOSUB 2100
270 '
280 'accept user input & echo on screen
290 '
300 INPUT"semi-major axis (in km)=",H
310 ER=H/6378.165
320 IF ER<1 THEN BEEP:LOCATE 2,2:PRINT"semi-major axis must
be > 6378.165 km":GOTO 260
330 '
340 'draw globe
350 '
360 GOSUB 1770
370 LOCATE 2,1:PRINT"                                    "
380 LOCATE 4,1:PRINT"a =":PRINT USING"#####.#";H
390 PRINT"     km
400 '
410 LOCATE 8,1:PRINT"        "
420 PRINT"        ":PRINT"        "
430 LOCATE 11,1:PRINT"        "
440 PRINT"        ":PRINT"        "
450 LOCATE 15,1:PRINT"        "
```

```
460 PRINT"          ":PRINT"         "
470 LOCATE 19,1:PRINT"         "
480 PRINT"          ":PRINT"         "
490 GOSUB 2100:INPUT"eccentricity=",E
500 LOCATE 8,1:PRINT"e =":PRINT USING".######";E
510 GOSUB 2100
520 INPUT"inclination (deg)=",I
530 LOCATE 11,1:PRINT"i =":PRINT USING"###.###";I
540 PRINT"    deg
550 GOSUB 2100
560 '
570 'if eccentricity = 0 then skip arg of per and asc node
580 '
590 IF E=0 THEN GOTO 710
600 '
610 INPUT"argument of perigee (deg)=",W
620 LOCATE 15,1:PRINT"w =":PRINT USING"###.###";W
630 PRINT"    deg
640 GOSUB 2100
650 INPUT"ascending node (deg)=",O1
660 LOCATE 19,1:PRINT CHR$(234)" =":PRINT USING"###.###";O1
670 PRINT"    deg
680 '
690 'turn the sphere for edge view
700 '
710 O1=O1+89
720 GOSUB 2100
730 '
740 'change to radians
750 '
760 W=W*TORAD
770 I=I*TORAD
780 O=O1*TORAD
790 '
800 'P Q R coordinate system
810 '
820 PX=COS(W)*COS(O)-SIN(W)*SIN(O)*COS(I)
830 PY=COS(W)*SIN(O)+SIN(W)*COS(O)*COS(I)
840 PZ=SIN(W)*SIN(I)
850 QX=-SIN(W)*COS(O)-COS(W)*SIN(O)*COS(I)
860 QY=-SIN(W)*SIN(O)+COS(W)*COS(O)*COS(I)
870 QZ=COS(W)*SIN(I)
880 '
890 'mean motion
900 '
910 N=K/ER^1.5
920 '
930 'period
940 '
950 P=TWOPI/N
960 '
970 GOSUB 2100
```

```
980 '
990 'main program loop
1000 '
1010 FOR TTT=0 TO (P+.1) STEP P/50
1020 '
1030 'kepler's equation
1040 '
1050 GOSUB 1640
1060 '
1070 'xw  yw   rw  coordinates
1080 '
1090 XW=ER*(COS(EN1)-E)
1100 YW=ER*SQR(1-E^2)*SIN(EN1)
1110 RW=ER*(1-E*COS(EN1))
1120 RX=RW*6378.165
1130 '
1140 'calculate velocity & N (rev/day)
1150 '
1160 VEL=SQR(398603.2*((2/RX)-(1/H)))
1170 N1=16.997*(1/ER)^1.5
1180 '
1190 'print vel, N, period, rad on screen
1200 '
1210 GOSUB 2140
1220 '
1230 'determine if orbit hits earth
1240 '
1250 IF RW <= 1 THEN LOCATE 3,1:PRINT"  Satellite will hit
earth (suborbital)":LOCATE 2,1
1260 '
1270 'x   y    z  coordinates
1280 '
1290 X=XW*PX+YW*QX
1300 Y=XW*PY+YW*QY
1310 Z=XW*PZ+YW*QZ
1320 R=SQR(X^2+Y^2+Z^2)
1330 Z1=Z*(20/19)
1340 'LOCATE 23,1:PRINT Y POINT(-X,Z1)
1350 '
1360 'check to see if orbit is behind earth
1370 '
1390 IF Y<0 AND POINT(-X,Z1)<>0 GOTO 1490
1410 '
1420 'plot orbit
1430 '
1440 CIRCLE(-X,Z1),RO/20,2
1450 PAINT(-X,Z1),2
1460 '
1470 'end of main loop
1480 '
1490 NEXT TTT
1500 '
```

```
1510 'pause and then give choices
1520 '
1530 FOR S=1 TO 1000:NEXT S
1540 LOCATE 1,1:PRINT"
                        "
1550 LOCATE 1,5:PRINT"another satellite at same altitude?
1560 X$=INPUT$(1)
1570 IF X$="n" THEN LOAD"pick",R
1580 IF X$="y" THEN A=0:I=0:E=0:W=0:01=0:GOTO 400 ELSE 1540
1590 '
1600 END
1610 '
1620 'Kepler's eqn subroutine
1630 '
1640 EEE=90
1650 EEERAD=EEE*(PI/180)
1660 M=N*TTT
1670 FOR I=1 TO 100
1680 IF ABS(EN1-EEERAD)<.000001 GOTO 1740
1690 EEERAD=EN1
1700 MN=EEERAD-(E*SIN(EEERAD))
1710 DM=1-(E*COS(EEERAD))
1720 EN1=EEERAD+((M-MN)/DM)
1730 NEXT I
1740 RETURN
1750 STOP
1760 '
1770 'routine to draw globe
1780 '
1790 XO=0:YO=0:RD=ER*2
1800 R=1
1810 A=PI
1820 COSA=COS(A):SINA=SIN(A)
1830 CLS
1840 WINDOW(XO-RD,YO-RD)-(XO+RD,YO+RD)
1850 CIRCLE(0,0),.99,1
1860 PAINT(0,0),1
1870 FOR S=0 TO PI STEP PI/6
1880 X=0
1890 Y=R*COSA*COS(S)
1900 RO=R*SIN(S)
1910 CIRCLE(X,Y),RO,3,PI,TWOPI,ASPO
1920 NEXT S
1930 FOR I=0 TO 6
1940 FOR J=0 TO 12
1950 T=I*PI/6
1960 S=J*PI/12
1970 COST=COS(T):SINT=SIN(T)
1980 COSS=COS(S):SINS=SIN(S)
1990 X=R*COST*SINS
2000 Y=(R*SINA*SINT*SINS+R*COSA*COSS)
2010 LINE -(X,Y),3
```

```
2020 PSET(X,Y),3
2030 NEXT J
2040 NEXT I
2050 RETURN
2060 STOP
2070 '
2080 'subroutine to print rad, vel, per, & N on screen
2090 '
2100 LOCATE 1,5
2110 PRINT"                                        "
2120 LOCATE 1,5
2130 RETURN
2140 '
2150 LOCATE 1,1:PRINT"Rad=
2160 LOCATE 1,5:PRINT USING"######.##";RX
2170 LOCATE 1,15:PRINT"km
2180 LOCATE 1,21:PRINT"Vel=
2190 LOCATE 1,25:PRINT USING"###.#####";VEL
2200 LOCATE 1,35:PRINT"km/sec
2210 LOCATE 2,1:PRINT"per=
2220 LOCATE 2,5:PRINT USING"######.##";P
2230 LOCATE 2,15:PRINT"min
2240 LOCATE 2,21:PRINT"  N=
2250 LOCATE 2,25:PRINT USING"###.####";N1
2260 LOCATE 2,34:PRINT"rev/day
2270 RETURN
2280 STOP
2290 '
2300 'introduction
2310 '
2320 LOCATE 5,18
2330 PRINT"ORBIT
2340 LOCATE 8,1
2350 PRINT"  This program allows you to enter
2360 PRINT"  five of the six classical orbital
2370 PRINT"  elements.  The sixth, time, is
2380 PRINT"  represented by starting the orbit
2390 PRINT"  at the perigee point and ending
2400 PRINT"  one complete orbit later.  You may
2410 PRINT"  then display another orbit of the
2420 PRINT"  same altitude over the first.
2430 LOCATE 17,8
2440 LINE(3,5)-(315,150),3,B
2450 INPUT"press enter to continue...",KAL
2460 CLS
2470 RETURN
2480 STOP
```

APPENDIX D

GROUND TRACK PROGRAM LISTING

```
10 'ground track
20 '
30 'set up the screen
40 '
50 KEY OFF:CLS:SCREEN 1,0
60 COLOR 9
70 '
80 'introduction
90 '
100 GOSUB 2690
110 '
120 'draw the map of the earth
130 '
140 GOSUB 2290
150 '
160 'constants
170 '
180 PI=3.141593
190 TWOPI=2*PI
200 TORAD=1.745329E-02
210 TODEG=57.2957795#
220 K=7.436575E-02:DTHETA=.004369
230 RE=6378.165
240 '
250 'change color of satellite tracks
260 '
270 KAL=KAL+1
280 IF KAL=1 THEN C=2
290 IF KAL=2 THEN C=1
300 IF KAL=3 THEN C=3
310 IF KAL=4 THEN C=2
320 IF KAL=5 THEN C=1
330 IF KAL=6 THEN C=3
340 IF KAL=7 THEN C=2
350 '
360 'input the orbital elements
370 '
380 LOCATE 1,1
390 PRINT"                                        "
400 LOCATE 1,1
410 INPUT"semi-major axis (in km)=",A
420 IF A<RE THEN BEEP:GOTO 380
430 LOCATE 6,1:PRINT"a =
440 PRINT USING"#####";A
```

```
450 PRINT"   km
460 A=A/RE
470 LOCATE 1,1
480 PRINT"                                        "
490 LOCATE 1,1
500 INPUT"eccentricity=",E
510 LOCATE 10,1:PRINT"e =
520 PRINT USING".##";E
530 LOCATE 1,1
540 PRINT"                                        "
550 LOCATE 1,1
560 INPUT"inclination (deg)=",I
570 LOCATE 13,1:PRINT"i =
580 PRINT USING"###";I
590 PRINT"deg
600 IF E=0 THEN O1=0:W=0:GOTO 750
610 LOCATE 1,1
620 PRINT"                                        "
630 LOCATE 1,1
640 INPUT"argument of perigee (deg)=",W
650 LOCATE 17,1:PRINT"w =
660 PRINT USING"###";W
670 PRINT"deg
680 LOCATE 1,1
690 PRINT"                                        "
700 LOCATE 1,1
710 INPUT"ascending node (deg)=",O1
720 LOCATE 21,1:PRINT CHR$(234)" =
730 PRINT USING"###";O1
740 PRINT"deg
750 LOCATE 1,1
760 PRINT"                                        "
770 LOCATE 1,1
780 INPUT"number of orbits=",NR
790 LOCATE 1,1:PRINT"                                        "
800 '
810 'change to radians
820 '
830 W=W*TORAD
840 I=I*TORAD
850 O=O1*TORAD
860 '
870 'change to P Q R coordinate system
880 '
890 PX=COS(W)*COS(O)-SIN(W)*SIN(O)*COS(I)
900 PY=COS(W)*SIN(O)+SIN(W)*COS(O)*COS(I)
910 PZ=SIN(W)*SIN(I)
920 QX=-SIN(W)*COS(O)-COS(W)*SIN(O)*COS(I)
930 QY=-SIN(W)*SIN(O)+COS(W)*COS(O)*COS(I)
940 QZ=COS(W)*SIN(I)
```

```
950 '
960 'calculate mean motion and period
970 '
980 N=K/A^1.5
990 P=TWOPI/N
1000 '
1010 'draw the specified number of ground tracks
1020 '
1030 FOR TTT=0 TO NR*(P+.1) STEP P/50
1040 '
1050 'goto Kepler's equation
1060 '
1070 GOSUB 2160
1080 '
1090 'change to xw  yw   rw  coordinates
1100 XW=A*(COS(EN1)-E)
1110 YW=A*SQR(1-E^2)*SIN(EN1)
1120 RW=A*(1-E*COS(EN1))
1130 '
1140 'is the semi-major axis smaller than the earth's
radius?
1150 '
1160  IF RW <= 1 THEN LOCATE 1,1:PRINT"  Satellite will  hit
earth (suborbital)":FOR CCC=1 TO 1000:NEXT CCC:GOTO 2040
1170 '
1180 'change to x   y    z  coordinates
1190 '
1200 X=XW*PX+YW*QX
1210 Y=XW*PY+YW*QY
1220 Z=XW*PZ+YW*QZ
1230 '
1240 'calculate the radius of the orbit
1250 '
1260 R=SQR(X^2+Y^2+Z^2)
1270 '
1280 'calculate the longitude and screen x-coordinate
1290 '
1300 ALPHA=ATN(Y/X)
1310 ALPHA=ALPHA-DTHETA*TTT+O
1320 '
1330 'keep the correct quadrant
1340 '
1350 IF X<0 AND Y>0 THEN ALPHA=ALPHA+PI
1360 IF X<0 AND Y<0 THEN ALPHA=ALPHA+PI
1370 IF X>0 AND Y<0 THEN ALPHA=ALPHA+TWOPI
1380 IF X>0 AND Y>0 THEN ALPHA=ALPHA+TWOPI
1390 IF X=0 AND Y<=0 THEN ALPHA=PI/2
1400 IF X=0 AND Y>0 THEN ALPHA=PI*1.5
1410 '
1420 'change to degrees
```

```
1430 '
1440 LONG=ALPHA*TODEG
1450 '
1460 'keep longitude between 0 & 360 degrees
1470 '
1480 IF LONG>360 THEN LONG=LONG-360
1490 IF LONG<0 THEN LONG=LONG+360
1500 '
1510 'calculate the latitude
1520 LT=Z/R
1530 '
1540 'keep the latitude between 0 & 90 degrees
1550 '
1560 IF LT>1 THEN LT=1
1570 IF LT<-1 THEN LT=-1
1580 LT=ATN(LT/SQR(-LT*LT+1))
1590 '
1600 'change latitude to degrees
1610 '
1620 LAT=LT*TODEG
1630 '
1640 'add the ascending node to the longitude
1650 '
1660 LONG=LONG-O1
1670 '
1680 'size the latitude and longitude to the screen
1690 '
1700 LNG=LONG*(30/32)
1710 LNG=LNG*(32/36)
1720 LNG=LNG+169
1730 IF LNG>319 THEN LNG=LNG-300
1740 IF LNG<19 THEN LNG=LNG+300
1750 '
1760 'plot the groundtrack on the map
1770 '
1780 CIRCLE(LNG,100-LAT),1,C
1790 PAINT(LNG,100-LAT),C
1800 '
1810 'display the time, latitude and longitude on the screen
1820 '
1830 LOCATE 1,1:PRINT"TIME=
1840 LOCATE 1,6
1850 PRINT USING"#####.##";TTT
1860 LOCATE 1,15:PRINT"min
1870 LOCATE 1,19:PRINT"LAT=
1880 LOCATE 1,23
1890 PRINT USING"###.##";LAT
1900 LOCATE 1,30:PRINT"LNG=
1910 LOCATE 1,35
1920 PRINT USING"###.##";LONG
```

```
1930 '
1940 'end of loop
1950 '
1960 NEXT TTT
1970 '
1980 'delay the last display
1990 '
2000 FOR CCC=1 TO 1000:NEXT CCC
2010 '
2020 'give option to do another satellite
2030 '
2040  LOCATE 1,1:PRINT"
":LOCATE 1,1
2050 PRINT"another satellite?
2060 FOR CCC=1 TO 1500:NEXT CCC
2070 LOCATE 1,1:PRINT"
":LOCATE 1,1
2080 X$=INPUT$(1)
2090 IF X$="y" THEN GOTO 2390
2100 IF X$="n" THEN LOAD"pick",R ELSE 2050
2110 '
2120 END
2130 '
2140 'loop to solve Kepler's equation
2150 '
2160 EEE=90
2170 EEERAD=EEE*(PI/180)
2180 M=N*TTT
2190 FOR I=1 TO 100
2200 IF ABS(EN1-EEERAD)<.000001 GOTO 2260
2210 EEERAD=EN1
2220 MN=EEERAD-(E*SIN(EEERAD))
2230 DM=1-(E*COS(EEERAD))
2240 EN1=EEERAD+((M-MN)/DM)
2250 NEXT I
2260 RETURN
2270 STOP
2280 '
2290 'subroutine to load map of the earth
2300 '
2310 DIM SCRN(4048),ATOS(6):LA=0
2320 DEF SEG:BLOAD"arryscrn",VARPTR(ATOS(0))
2330 PUTSCRN=VARPTR(ATOS(0)):LA=VARPTR(SCRN(0))
2340 BLOAD"picture",LA
2350 CALL PUTSCRN(SCRN(0))
2360 RETURN
2370 STOP
2380 '
2390 'erase old elements & redraw grid
2400 '
```

```
2410 LOCATE 6,1
2420 PRINT"   "
2430 PRINT"      "
2440 PRINT"      "
2450 LOCATE 10,1
2460 PRINT"   "
2470 PRINT"   "
2480 PRINT"   "
2490 LOCATE 13,1
2500 PRINT"   "
2510 PRINT"   "
2520 PRINT"   "
2530 LOCATE 17,1
2540 PRINT"   "
2550 PRINT"   "
2560 PRINT"   "
2570 LOCATE 21,1
2580 PRINT"   "
2590 PRINT"   "
2600 PRINT"   "
2610 LINE(19,10)-(19,190),1
2620 LINE(19,40)-(25,40),1
2630 LINE(19,100)-(25,100),3
2640 LINE(19,130)-(25,130),1
2650 LINE(19,160)-(25,160),1
2660 GOTO 270
2670 STOP
2680 '
2690 'introduction
2700 '
2710 LOCATE 5,13
2720 PRINT"GROUND TRACK
2730 LOCATE 8,1
2740 PRINT"  This program plots the ground track
2750 PRINT"  of an orbit on a cartesian map of
2760 PRINT"  the earth.  Each grid is 30 degrees
2770 PRINT"  by 30 degrees.  The equator and
2780 PRINT"  prime meridan are outlined in
2790 PRINT"  white.
2800 PRINT
2810 LINE(3,5)-(315,150),3,B
2820 RETURN
2830 STOP
```

APPENDIX E

GEOSYNC PROGRAM LISTING

```
10 'geosync
20 '
30 'set up screen
40 '
50 CLS:KEY OFF:SCREEN 1,0
60 '
70 PI=3.1415927#
80 '
90 'introduction and draw map
100 '
110 GOSUB 1120
120 GOSUB 1020
130 '
140 'accept and echo to screen user input
150 '
160 LOCATE 1,1
170 INPUT"inclination angle in degrees = ",III
180 IF III>90 OR III<0 THEN BEEP:GOTO 140
190 LOCATE 1,1:PRINT"                                          "
200 LOCATE 1,1
210 INPUT"longitude (0 to 360 deg)=",LG
220 IF LG>360 OR LG<0 THEN BEEP:GOTO 200
230 XXX=LG*(30/36)
240 LOCATE 1,1
250 PRINT"inclination="III"deg  longitude="LG"deg
260 III=III*PI/180
270 '
280 'calculations for rotation of the earth
290 '
300 WE=2*PI/24
310 WO=WE
320 '
330 'main program loop
340 '
350 FOR T=0 TO 18 STEP .25
360 Y=SIN(III)*SIN(WO*T)
370 '
380 'calculate latitude
390 '
400 IF Y=1 THEN Y=.999999
410 IF Y=-1 THEN Y=-.999999
420 LAT=ATN(Y/SQR(-Y*Y+1))
430 '
440 'calculate longitude
450 '
460 LONG=ATN(COS(III)*TAN(WO*T))-WE*T
470 IF LONG < -PI/2  THEN LONG = LONG + PI
480 IF LONG < 0 AND LAT <= -III THEN GOTO 710
```

```
490 '
500 'change to degrees
510 '
520 LAT=LAT*180/PI
530 LONG=LONG*180*PI
540 '
550 'size to and keep on screen
560 '
570  LONG=LONG*(1/12)
580 LONG=LONG+169+XXX
590 IF LONG<19 THEN LONG=LONG+300
600 IF LONG>319 THEN LONG=LONG-300
610 '
620 'plot ground track
630 '
640 CIRCLE(LONG,  100-LAT),1,2
650 PAINT(LONG,  100-LAT),2
660 '
670 'end of loop
680 '
690 NEXT T
700 '
710 '4th quadrant fix
720 '
730 FOR T=18 TO 24 STEP .25
740 Y=SIN(III)*SIN(WO*T)
750 IF Y=1 THEN Y=.999999
760 IF Y=-1 THEN Y=-.999999
770 LAT=ATN(Y/SQR(-Y*Y+1))
780 LONG=ATN(COS(III)*TAN(WO*T))-WE*T
790  LONG=LONG+(2*PI)
800 LAT=LAT*180/PI
810 LONG=LONG*180*PI
820  LONG=LONG*(1/12)
830 LONG=LONG+169+XXX
840 IF LONG<19 THEN LONG=LONG+300
850 IF LONG>319 THEN LONG=LONG-300
860 CIRCLE(LONG,  100-LAT),1,2
870 PAINT(LONG,  100-LAT),2
880 NEXT T
890 '
900 '
910 'choice of another satellite
920 '
930 LOCATE  1,1:PRINT"
":LOCATE 1,1
940 PRINT"another satellite?
950 FOR CCC=1 TO 1500:NEXT CCC
960 LOCATE 1,1:PRINT"
":LOCATE 1,1
970  X$=INPUT$(1)
980 IF X$="y" THEN GOTO 140
990 IF X$="n" THEN LOAD"pick",R ELSE 940
```

```
1000 STOP
1010 '
1020 'subroutine to load map of the earth
1030 '
1040 DIM SCRN(4048),ATOS(6):LA=0
1050 DEF SEG:BLOAD"arryscrn",VARPTR(ATOS(0))
1060 PUTSCRN=VARPTR(ATOS(0)):LA=VARPTR(SCRN(0))
1070 BLOAD"picture",LA
1080 CALL PUTSCRN(SCRN(0))
1090 RETURN
1100 STOP
1110 '
1120 'introduction
1130 '
1140 LOCATE 5,17
1150 PRINT"GEOSYNC
1160 LOCATE 8,1
1170 PRINT"  A geosynchronous orbit is a circular
1180 PRINT"  orbit located about 5.6 earth radii,
1190 PRINT"  19,300 nm, or 35,750 km above the
1200 PRINT"  earth.  Its period is 23 hr 56 min.
1210 PRINT
1220 PRINT"  If a geosynchronous orbit has an
1230 PRINT"  inclination of 0 degrees, it is
1240 PRINT"  considered to be geostationary.
1250 LINE(3,5)-(315,150),3,B
1260 RETURN
1270 STOP
```

APPENDIX F

SUGGESTED INPUT EXAMPLES

To start the programs, insert the disk in the "A" or default drive and turn on the computer. The program INTRO will start with the title page appearing on the screen. Press the enter key to read the program descriptions and then press enter again to arrive at the menu. To run a program, type the first two letters of the program name on the keyboard and the program will begin.

The following input is suggested to familiarize the user with the programs:

1. INITIAL CONDITIONS

When the program asks for the initial altitude, enter "0".

When the program asks for the flight path angle, enter "0".

When the program asks whether you want to enter the initial speed or apogee altitude, type "v" for speed and then enter "9".

After the program plots the orbit and asks if you want another satellite, type "n" for no. The program will ask if you want to see orbital data. Type "y" for yes.

The program will then ask if you want to try another satellite. Type "n" for no and the program returns you to the menu.

2. ORBIT

When the program asks for the semi-major axis, enter "8000".

The program then draws the sphere representing the earth. It will then ask for the eccentricity. Enter ".00001".

The inclination is then requested. Enter "45".

Enter "30" when the argument of perigee is asked for.

The last input is the ascending node. Enter "60".

The program then will draw the orbit around the sphere. When finished, the option to draw another orbit over the first is given. Type "n" for no and to return to the menu.

3. GROUND TRACK

This program has the same input as the previous program. Enter the same data to see what the ground track of the orbit looks like.

Another interesting orbit can be plotted by entering the following data:

semi-major axis	=	26620 km
eccentricity	=	.76
inclination	=	65 degrees
argument of perigee	=	270 degrees
ascending node	=	85 degrees
number of orbits	=	2

4. GEOSYNC

There are only two inputs for this program.

Enter "45" for inclination and "60" for longitude.

Another interesting orbit can be obtained by entering "90" for inclination and either "0" or "180" for longitude.

LIST OF REFERENCES

1. Sterne, T. E., "Celestial Mechanics of Artificial Satellites," Sky & Telescope, 15 October 1957.

2. Harris, D., Computer Graphics and Applications, Chapman and Hall, 1984.

3. Waite, M. and Morgan, C. L., Graphics Primer for the IBM PC, Osborne/McGraw-Hill, 1983.

4. Bate, R. R., Mueller, D. D., and White, J. E., Fundamentals of Astrodynamics, Dover, 1971.

5. Eisele, J. A. and Nichols, S. A., Orbital Mechanics of General-Coverage Satellites, Naval Research Laboratory, 1976.

6. Escobal, P. R., Methods of Orbit Determination, Wiley, 1965.

7. Nelson, W. C. and Loft, E. E., Space Mechanics, Prentice-Hall, 1962.

8. Kaplan, M. H., Modern Spacecraft Dynamics & Control, John Wiley & Sons, 1976.

9. Fuhs, A. E., AE 4791/4792 Spacecraft Design, class notes presented at the Naval Postgraduate School, 1985.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Lt Col J. Malokas, Code 39 Naval Postgraduate School Monterey, California 93943	1
4.	Space Academic Group, Code 72 Naval Postgraduate School Monterey, California 93943	1
5.	Professor G.L. Swafford, Code 61So Naval Postgraduate School Monterey, California 93943	3
6.	Professor Otto Heinz, Code 61Hz Naval Postgraduate School Monterey, California 93943	2
7.	Professor Allen E. Fuhs, Code 72Fu Naval Postgraduate School Monterey, California 93943	1
8.	Dean J.N. Dyer, Code 06 Naval Postgraduate School Monterey, California 93943	1
9.	Commander, Naval Space Command Room 17, Bldg 183 Dahlgren, Virginia 22448	1
10.	FL2508, Space Library HQ AFSPACECOM/MPSL Peterson AFB, Colorado 80914	1
11.	Commander, USSPACECOM/UDE Peterson AFB, Colorado 80914-5001	1

12.	Mr. C.W. Langdorf Box 83, Unit CB APO NY 09154	1
13.	CPT Kim A. Langdorf HQ USSPACECOM/J4/6S Peterson AFB, Colorado 80914-5001	5